Extended Abstract: Image Matching for Branding Phishing Kit Images

Chengcui Zhang, Rajan Kumar Kharel, Jason Britt, Song Gao
Department of Computer & Inf. Sciences
University of Alabama at Birmingham
Birmingham, AL 35294

Abstract

Phishing websites attempt to convince people to deliver their passwords, user IDs and other sensitive information by imitating legitimate websites such as banks, product vendors, or service providers. Using a phishing kit (compressed file) is a preferred way of creating phishing websites as it allows fast deployment of a phishing site. A kit may contain one or more images that are similar to the targeted brand such as a bank logo or a product trademark. In this paper, we explore the feasibility of using automatic image matching techniques to identify a kit's targeted brand. To this end, we evaluate the ability of image matching algorithms to correctly identify a pool of images from suspected kits. Four image-matching algorithms are evaluated based on their accuracy of branding images extracted from suspected phish kits.

1. Introduction

A phishing website usually selects a particular target (e.g., a bank), and incorporates one or more images that are similar to a targeted brand whether the image is located on the same domain as the phish or a non-local domain. One common method of distributing phishing websites is to use a "phishing kit" or kit, which is a compressed file folder containing all files and directory structures necessary to create a phishing website. A kit is often used repeatedly by a single criminal or criminal group and is a preferred way of creating phishing websites. The kit contains any email address receiving the phished credentials, which can be important during investigations. When identifying a phishing kit's brand, it cannot always be assumed that the phishing kit has the same brand as the phishing website where it was found. Multiple phishing websites can be setup on the same domain and unused kits can be located on active phishing domains. A kit's brand is useful when alerting the organization being targeted or allowing brand specific investigations. Even though the identification can be accomplished manually it is time consuming and unfeasible for the UAB Kit Data Mine [4], given its size. Phishing kits often incorporate images that are similar to the targeted brand. Finding these brand relevant images and labeling them may lead to automated methods to brand phishing kits. Simple hash matching techniques are limited because it is easy to alter an image's hash and not its meaning. More robust automated methods are needed to help reduce or eliminate manual effort. The rest of this paper explores the ability of image matching techniques to correctly identify image files associated with a brand. Four image-matching algorithms GCH, LCH, LCH+, and LCH++ are explored.

2. Related Work

Image analysis has been applied in phishing research to differentiate between phishing and non-phishing websites. Dunlop et al. [1] use Optical Character Recognition (OCR) to extract text from an image generated by converting a rendered web page, and applying the Google page rank algorithm on the text to determine if the corresponding website appears in the top search results. If the website does not appear in the top four search results, the web page is considered a phishing page. Fu et al. [2] present a phishing web page detection algorithm, which uses Earth Mover's Distance (EMD) to measure Web page visual similarity. A threshold is calculated for each protected Web page using supervised learning. A web page is classified as a phishing page if its EMD-based similarity exceeds the threshold of a protected web page. Cordero et al. [3] propose a system that detects phish by using a computer vision based approach on rendered website images. A joint histogram based on color and edge density features is computed for each image, resulting in a 256-feature vector. Principle Component Analysis (PCA) is applied to project the 256-*d* dataset into a 4-*d* space. Support vector machine (SVM), Naïve Bayes, and K-Nearest Neighbors (KNN) are used respectively to build classifiers based on the 4-*d* training dataset for testing a rendered website images. However, to our best knowledge, our work is the first of its kind that explores the ability to brand images contained in phishing kits.

3.0. Global Color Histogram (GCH) & Local Color Histogram (LCH)

Each pixel in an image is represented by a 6-bit color code [5] formed by taking the two most significant bits

from each 8-bit R, G, and B channel. Each 6-bit pixel representation lies in the value range of [0, 63]. Therefore, a 64-bin histogram is created for each image. Global Color Histogram (GCH) is a standard image analysis technique [6] to represent the color distribution in an image. Each bin in a histogram represents the percentage (relative representation) of pixels that have color values within the corresponding fixed color range. This representation is robust to scaling and allows fair comparison between two images of different size. Local Color Histogram (LCH) is generated by dividing the image into $M \times N$ grid cells (e.g. 3×3) and calculating a color histogram on each cell. The similarity between two images by using LCH features is calculated as the sum or average similarity of corresponding cell pairs. GCH is not sensitive to the location of the object-of-interest in an image, since location change alone will not affect the color distribution that much. However, two visually different images may have very similar color histograms, contributing many false positives in matching images, which may defeat the purpose of using automatic image matching for branding kit images. On the other hand, LCH incorporates spatial distribution of colors and produces many fewer false positives.

3.2. Local Color Histogram with Preprocessing (LCH+, LCH++)

We explore several preprocessing techniques. The first preprocessing technique is a dimensional constraint. If the difference in the aspect ratios of two images is greater than a threshold value, the two images are considered unmatched and will not be processed further. A training set consisting of pairs of matched brand images is used to determine this threshold. Specifically, the threshold value is set to be the maximum aspect ratio (height/width) difference among all pairs of matched images.

(a)

(b)

Figure 1: (a) The original image MBB (dashed box) (b) Extracted foreground (left) mask (right)

The second preprocessing technique is background color removal. Some brand images (e.g. Figure 1(a)) have a large background area with almost pure color which is brand-irrelevant and introduces noises into LCH-based image matching. For such an image, the most dominant bin of its global color histogram represents the background, and the pixels falling into that bin will be excluded from any subsequent calculation of local histograms. Images with the same brand almost always have a similar background color. In this work, two criteria are used to determine whether two images to be compared should have their background pixels removed: 1) the difference between the first two dominant bins of each of these two histograms must be greater than 0.5, indicating the possible existence of a large pure-colored background, and 2) The first dominant bins of both images must correspond to the same color code. The backgrounds pixels of both images are then removed (Figure 1(b)). The third preprocessing technique is the Minimum Bounding Box (MBB) technique. The MBB technique is used to extract a bounding box of the foreground area. Figure 1 shows an example of foreground MBB extraction. Figure 2(a) shows images that have similar background; Figure 2(b) shows the images after the removal of the background pixels.

(a)

(b)

Figure 2: Background removal examples

LCH+ employs the dimensional constraint and background removal steps and then applies the LCH-based comparison. LCH++ uses the dimensional constraint, MBB, and background removal steps and then applies the LCH-based comparison.

3.3. Experimental Dataset

Our image dataset consists of images found in suspected kits in the Kit Data Mine [4] collected between 2010-07-16 and 2012-08-23. All MD5 distinct images are manually labeled as 'brand' with a specific brand name or 'non-brand' with a label of "general image". There are 215 brand relevant images and 9,915 general images. There are 42 brands represented in the 215 brand relevant images. The first chronologically occurring 109 brand relevant images were chosen as the training set, roughly splitting the time period in half. The other 106 brand relevant images and the 9,915 general images make up the test set. The brand relevant images in the training and test sets have representatives for each of the 42 brands.

3.4. Image Matching

Each of the four algorithms is applied within the same four-step clustering process. First, all pure color images that have one single non-zero bin in the corresponding

color code histogram are excluded, as they are not brand images. Second, a color code histogram is generated for all images (training and testing) based on the adopted algorithm (one of the abovementioned). Different similarity measures have been suggested to compare two color histograms [7,8]. In our case, histogram intersection (HI) is used to calculate the distance (D) between two color histograms since it is known to outperform Euclidean distance in image matching. The distance between two images I_i and I_j is defined in (Eq1).

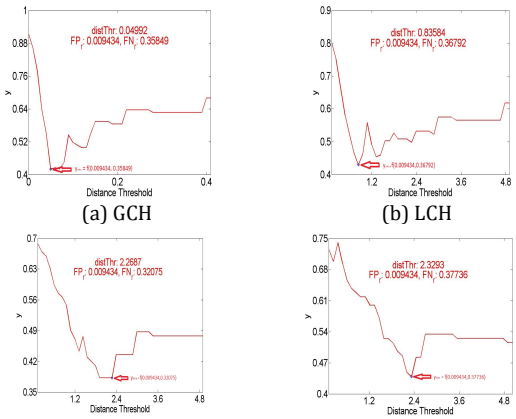
$$D(I_i, I_j) = 1 - \sum_{i=1}^{n} \min\left(H_i(b), H_i(b)\right) \qquad \text{(Eq 1)}$$

where H_i and H_j are the color-code histograms for images I_i and I_j, respectively; b is the bin index; n = 64 is the total number of bins in the color-code histograms. Third, the 109 training images are set as cluster seeds. Fourth, each image in the test set (106 brand relevant images and the 9,915 general images) is assigned to the corresponding cluster seed with the lowest distance that is below a minimum threshold. A true positive (TP) is a brand image assigned to its correct brand cluster. A false positive (FP) is a general image assigned to a brand cluster, or a brand image assigned to an incorrect brand cluster. A false negative (FN) is a brand image not assigned to a brand cluster. A true negative (TN) is a general image not assigned to any cluster.

3.5. Threshold Selection

In this section, we describe our strategy for selecting the minimum distance threshold for each algorithm. The main goal is to link a brand image to its correct brand cluster. Therefore, we focus primarily on keeping the false positive rate as low as possible, while maintaining a reasonably low false negative rate (e.g. <40%). The 106 brand relevant images from the testing set are used to determine an appropriate minimum distance threshold for each algorithm using the following three steps: 1) The maximum and minimum distance values between all testing and training images pairs are found. 2) 42 distance thresholds are evenly set between the maximum and minimum values. False negative rates (FN_r) and false positive rates (FP_r) are calculated for each distance threshold. 3) The 'optimal' threshold is determined when an objective function (Eq 2) reaches a minimum. w represents the FP_r weight, which is always greater than one, meaning that FP_r is w times as important as FN_r. We choose to calculate w using the ratio of test set general images (9,915) to test set brand images (106) reduced by one order of magnitude (93.538/10=9.3538 in our case). Figure 3 shows the variation of y values with respect to the variation of distance thresholds.

$$y = (w \times FP_r) + FN_r \qquad \text{(Eq 2)}$$



(a) GCH (b) LCH (c) LCH+ (d) LCH++

Figure 3: Threshold selection for each algorithm

4. Results and Discussions

The dataset described in Sections 3.3 is used to evaluate the performance of the four algorithms. Accuracy, as defined by (Eq 3), is used to evaluate the effectiveness of all four algorithms. Table 1 shows the accuracy values for the four algorithms.

$$Accuracy = \frac{(TP+TN)}{Total\ \#\ of\ testing\ images} \qquad \text{(Eq 3)}$$

Table 1: Evaluation results (test images: 10,021)

Algorithm	GCH	LCH	LCH+	LCH++
TP #	67	62	71	65
TN #	8,945	9,046	9,822	9,864
FP #	971	870	94	52
FN #	38	43	34	40
Accuracy (%)	89.93	90.88	98.72	**99.08**

In almost all the experiments, 99% of the false positives are caused by general images wrongfully assigned to brand clusters, which are partially attributed to the threshold selection that focuses on minimizing false positives among brand images. While LCH++ yields the overall lowest FP, the cause for all the false negatives is that there is no matching/similar image in the training set. In experimenting with all the four algorithms, the number of FPs is much larger than the number of FNs, due to there being only 106 brand test images and a much larger pool of general test images (9,915 of them). LCH++ has the best overall accuracy. LCH+ also produces comparable results, yielding accuracy only ~0.6% worse than that of LCH++, which corresponds to approximately 30 more miss-classified test

images. Both LCH+ and LCH++ yield a FP rate significantly lower than that of GCH and LCH. This demonstrates that the preprocessing techniques, including dimensional scaling, background removal, and foreground extraction, are all helpful in further improving the accuracy.

(a) (b) (c)

Figure 4: (a) FP from GCH (b) FN from LCH (c) A FP for GCH, LCH, LCH+, LCH++

Some examples of FPs and FNs are given in Figure 4. In Figure 4(a), a general image (the bottom image) is incorrectly assigned to a "Santander" brand cluster (the top image) when the GCH-based algorithm is used for matching. In contrast, the LCH-based algorithms, including LCH+ and LCH++, can correctly distinguish between them, as both take into consideration the local spatial color distribution, and thereby reducing the false positives. Only occasionally GCH-based algorithm performed better than LCH-based algorithms. The two "VISA" brand images shown in Figure 4(b) have a similar global color distribution and can be correctly linked by GCH-based algorithm, while LCH-based algorithms failed because of the difference in local color distributions.

The example in Figure 4(c) exposes the limitation of image matching algorithms based on color histogram features. The two images in this figure have very similar global color distributions, local color distributions as well as foreground color distributions. Therefore, all the four algorithms fail to differentiate between these two, yielding a false positive.

5. Conclusions and Future Work

We develop four image-matching algorithms based on color histogram features for automatically identifying brand images from phishing kit images. GCH-based algorithm is not sensitive to spatial rearrangement of color pixels (e.g. location change of the foreground logo) because color features are extracted from the entire image without considering their spatial distribution. Such characteristics cause more false positives than LCH-based algorithms where two images have similar global color distribution but different visual content. LCH-based algorithms extract color features within each local area/cell of an image, thereby producing less false positives without necessarily incurring more false negatives. The false positive rate can be further lowered by utilizing our proposed preprocessing techniques, including dimensional scaling, background removal, and foreground extraction.

Exploration of other visual features such as textures and shapes will be part of our future work. The screenshot images of phishing websites may also be considered for branding. To further reduce the computational cost incurred by exhaustive matching, a multi-dimension index mechanism will be incorporated to index brand images in the knowledge base. A learning mechanism will be put in place to continuously update the knowledge base of brand image clusters. Finally methods to utilize the branded images for kit branding will be evaluated.

6. References

[1] M. Dunlop, S. Groat and D. Shelly, "GoldPhish: Using Images for Content-Based Phishing Analysis," in the 5th Intl. Conf. on Internet Monitoring and Protection, 2010.

[2] A. Y. Fu and X. Deng, "Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover's Distance (EMD)," in IEEE TRANS. on Dependable and Secure Computing, 2006.

[3] A. Cordero and T. Blain, "Catching Phish: Detecting Phishing Attacks from Rendered Website Images," in Proc. of the 16th Intl. Conf. on World Wide Web, 2007.

[4] UAB Kit Data Mine, www.cis.uab.edu/PhishOps

[5] C. Zhang, W.-B. Chen, et al., "A Multimodal Data Mining Framework for Revealing Common Sources of Spam Images," Journal of Multimedia, Vol. 4, No. 5, Oct. 2009, pp. 313-320.

[6] M. J. Swain and D. H. Ballard, "Color Indexing," in Intl. J. of Computer Vision, 1991.

[7] P. Howarth and S. Ruger, "Fractional distance measures for content-based image retrieval," in European Conf. on Information Retrieval, Santiago de Compostela, Spain, 2005.

[8] Y. Rubner, C. Tomasi and L. Guibas, "The Earth Mover's Distance as a metric for image," Technical Report STAN-CS-TN-98-86, Computer Science Department, Stanford, 1998.