Framework for Multi-Features Based Web Harmful Information Identification

Dhanya V
Dept. of Computer Science
College of Engineering Perumon
Kerala, India

Praveen K Wilson
Assistant Professor
Dept. of Information Technology
College of Engineering Perumon
Kerala, India

Abstract

Phishing is an attempt to obtain sensitive and personal information by masquerading as a legitimate entity in some form of electronic communication. Phishing attacks have deceived many users by imitating websites and stealing personal information and/or financial data. This project proposes a novel framework using Bayesian approach for phishing web page detection. This model takes into account textual and visual contents to measure the similarity between the protected web page and suspicious web pages. A text classifier, an image classifier, and an algorithm fusing the results from classifiers are introduced. An outstanding feature of this project is the exploration of a Bayesian model to estimate the matching threshold. This is required for determining the class of the web page and identifying whether the web page is phishing or not. In the text classifier, the naive Bayes rule is used to calculate the probability that a web page is phishing. In the image classifier, the visual similarity is measured, and Bayesian model is designed to determine the threshold. In the data fusion algorithm, the weighting approach is used to synthesize the classification results from textual and visual classifiers.

Index Terms— phishing, Bayesian model, classification, weighting approach, data fusion

1. Introduction

World Wide Web (WWW) service started in the year 1991 and is gaining popularity day by day.The number of users using the Internet is rapidly increasing. The Web has become an indispensable global platform that glues together daily communication, sharing, trading, collaboration, and service delivery. Web users often store and manage critical information that attracts cybercriminals who misuse the web and the internet to exploit vulnerabilities for illegitimate benefits. Phishing attacks are one of the most crucial modern security threats in the current World Wide Web. Phishers often exploit users' trust on the appearance of a site by using web pages that are visually similar to an authentic site. According to the Anti-Phishing Working Group (APWG), there were at least 67677 phishing attacks in the last six months of 2010.Automatically detecting phishing web pages has attracted much attention from security and software providers, financial institutions, to academic researchers. Methods for detecting phishing web pages can be classified into industrial toolbar based anti-phishing, user-interface-based anti-phishing, and web page content-based anti-phishing. To date, techniques for phishing detection used by the industry mainly include authentication, filtering, attack tracing and analyzing, phishing report generating, and network law enforcement. These anti-phishing internet services are built into e-mail servers and web browsers and available as web browser toolbars (e.g., SpoofGuard Toolbar1 , TrustWatch Toolbar, and Netcraft Anti-Phishing Toolbar). These industrial services, however, do not efficiently thwart all phishing attacks. Wu *et al.* [1] conducted thorough study and analysis on the effectiveness of anti-phishing toolbars, which consist of three security toolbars and other mostly used browser security indicators. The study indicates that all examined toolbars in [1] were ineffective to prevent web pages from phishing attacks. Cranor *et al*. [2] performed another study on an evaluation of 10 anti-phishing tools. They indicated that only one tool could consistently detect more than 60% of phishing web sites without a high rate of false positives, whilst four tools were not able to recognize 50% of the tested web sites.

With respect to previous work, we clarify that our approach is most related to the content-based approaches such as CANTINA [3], visual similarity-based methods [4],[5]–[7], and machine learning techniques [8]–[10]. But the anti-phishing model proposed here is considerably different. In CANTINA [3], the formation of lexical signature is only based on several unique terms extracted from a given web page. The lexical signature is subsequently applied to the search engine. The generated lexical signature for the given web page matches with the domain name of billions of online web pages. The classification is based on the measurement from the Page Rank [11] assumption. In our detection framework, the existence of a protected web page, i.e., a legitimate web page, needs to be determined in the first place. Thus, based on the statistics from the attack historical data of the protected web page, the system classifies a given web page into the corresponding category, i.e., either phishing or normal. In addition, we include the conditional probabilities of all words, while CANTINA essentially relies on identifying the most unique terms. Compared with the detection methods of [3], [5]–[7], we extend these methods into a hybrid antiphishing framework, by taking additional content into account. Currently, we only include textual content as the additional content. Other surface level characteristics such as hyperlinks can also be easily combined into this framework. Here we use the EMD method [4] to assess the visual similarity of web pages. The visual similarity measurements of [5]–[7]can also be easily used in this framework. In the text classifier, we at present use the naive Bayes rule to classify web pages. we determine the threshold used in classifiers by using the Bayesian approach. Our proposed fusion algorithm based on the Weighting approach is also novel for phishing detection.

The rest of this paper is organized as follows: Section 2 describes the previous work in related domains. Section 3 presents the proposed framework. In Section 4, we introduce the text classifier based on the textual content of web pages. In Section 5, we introduce the image classifier . In Section 6, we introduce the Bayesian approach to estimate the threshold required in either the text classifier or the image classifier. In Section 7, we propose a novel fusion algorithm to combine the results from both classifiers. Finally, a short discussion on conclusions and future study are provide in Section 8.

2. Related work

Current phishing detection approaches fall into three main categories: (1) Non-content based approaches that do not use content of the site to classify it as authentic or phishing, (2) Content based approaches that use site contents to catch phishing, and (3) Visual similarity based approaches that identify phishing using their visual similarity with known sites. Other anti-phishing approaches include detecting phishing emails (rather than sites) and educating users about phishing attacks and human detection methods.

2.1. Non-content based approaches

Non-content based approaches include URL and host information based classification of phishing sites, blacklisting and whitelisting methods.

In URL based schemes , URLs are classified based on both lexical and host features. Lexical features describe lexical patterns of malicious URLs. These include features such as length of the URL, the number of dots, special characters it contains. Host features of the URL include properties of IP address, the owner of the site, DNS properties such as TTL, and geographical location . Using these features, a matrix is built and run through multiple classification algorithms.

In Blacklisting approaches, users report or companies seek and detect phishing sites' URLs which are stored in a database. Most commercial toolbars Netcraft , Internet explorer 7, CallingID Toolbar, EarthLink Toolbar , Cloudmark Anti- Fraud Toolbar , GeoTrust TrustWatch Toolbar , Netscape Browser 8.1 use this approach. But as most phishing sites are short-lived, last less than 20 hours , or change URLs frequently (fast-flux), the URL blacklisting approach fails to detect most phishing attacks. Furthermore, a blacklisting approach will fail to detect an attack that is targeted to a particular user ("spearphishing"), particularly those that target lucrative but not widely used sites such as company intranets.

Whitelisting approaches seek to detect known good sites, but a user must remember to check the interface every time he visits any site. Some whitelisting approaches use server side validation to add additional authentication metrics(beyond SSL) to client browsers as a proof of its benign nature,for example, Dynamic security skins , TrustBar SRD.

2.2. Content based approaches

In content based approach, phishing attacks are detected by examining site contents. Features used in this approach include spelling errors, source of the images, links, password fields, embedded links, etc. along with URL and host based features. SpoofGuard and CANTINA are two such approaches. Google's anti-phishing filter detects phishing and malware by examining page URL, page rank, WHOIS information and contents of a page including HTML, javascript, images, iframe, etc.. The classifier is regularly re-trained with new phishing sites to pick up new trends in phishing. This classifier has high accuracy but is currently used offline as it takes 76 seconds on average to detect phishing. Several researchers explored fingerprinting and fuzzy logic based approaches that use a series of (exact) hashes of websites to identify phishing sites. Our experimentation with a fuzzy hashing based approach suggested that thisapproach can detect current attacks, but can be easily circumvented by restructuring HTML elements without changing the appearance of the site .

2.3.Visual similarity based phishing detection

Chen et al. used screenshot of webpages to detect phishing sites . They used Contrast Context Histogram (CCH) to describe the images of webpages and k-mean algorithm to cluster nearest keypoints. Finally euclidean distance between two descriptors is used to find matching between two sites. analyzing screenshot is too slow to be used for online phishing detection. Fu et al. used Earth Mover's Distance (EMD) to compare low resolution screen capture of a webpage . Images of webpages are represented using color of a pixel in the image (alpha, red, green, and blue) and the centroid of its position distribution in the image. They used machine learning to select different threshold suitable for different webpages.

3. Overview of work done

3.1 Content representations

To summarize the whole content information of a web page, we divide the content representation into three categories.

1) Surface level content: “Surface level content” here is defined as the characteristics that are used by the users to access to a web page or to connect to other web pages. Such surface-level content consists of the domain name, URL, and hyperlinks which are involved in a given web page.

2) Textual content: “Textual content” in this paper is defined as the terms or words that appear in a given web page, except for the stop words.

3) Visual content: “Visual content” refers to the characteristics with respect to the overall style, the layout, and the block regions including the logos, images, and forms.

The proposed anti-phishing approach contains the following components.

1) A Text classifier using the naive Bayes rules to handle the text content extracted from a given web page.

2) An Image classifier using the EMD similarity method [4] to handle the pixel level content of a given web page that has been transformed into an image.

3) A Bayesian approach to estimate the threshold used in classifiers through offline training.

4) A data fusion algorithm to combine the results from the text classifier and the image classifier.

3.2 Overview of the system framework

The system includes a training section, which is to estimate the statistics of historical data (i.e., web page training set), and a testing section, which is to examine the incoming testing web pages. The statistics of the web page training set consists of the probabilities that a textual web page belongs to the categories (i.e., phishing and normal), the matching thresholds of classifiers, and the posterior probability of data fusion. Through the preprocessing, content representations, i.e., textual and visual, are rapidly extracted from a given testing web page. The text classifier is used to classify the given web page into the corresponding category based on the textual features. The image classifier is used to classify the given web page into the corresponding category based on the visual

content. Then the fusion algorithm is used to combine the detection results delivered by the two classifiers.
The detection results are eventually transmitted to the online users or the web browsers.

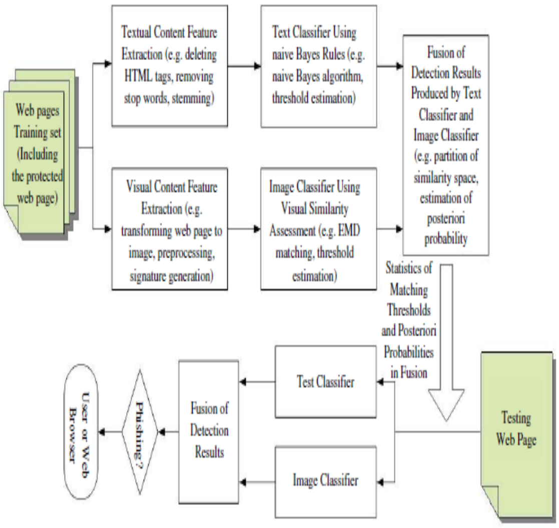
Web pages Training set (Including the protected web page)
Textual Content Feature Extraction (e.g. deleting HTML tags, removing stop words, stemming)
Text Classifier Using naive Bayes Rules (e.g. naive Bayes algorithm, threshold estimation)
Visual Content Feature Extraction (e.g. transforming web page to image, preprocessing, signature generation)
Image Classifier Using Visual Similarity Assessment (e.g. EMD matching, threshold estimation)
Fusion of Detection Results Produced by Text Classifier and Image Classifier (e.g. partition of similarity space, estimation of posteriori probability
Statistics of Matching Thresholds and Posteriori Probabilities in Fusion
Testing Web Page
Test Classifier
Image Classifier
Fusion of Detection Results
Phishing?
User or Web Browser

Figure 1. System architecture

4 .Text classifier

4.1. Preprocessing

The web content is semi structured an contains formatting information in form of HTML tags. First all HTML tags are removed from the web pages, including punctuation marks. Common features that are part of every web site were considered as stop features(such as the word 'a', 'the' etc.) The next step is to remove stop words as they are common to all documents and does not contribute much in searching. Since some words carry similar meanings but in different grammatically form (such as "bank" and "banks"), therefore it is needed to combine them into one attribute. In most cases a stemming algorithm is applied to reduce words to their basic stem.

4.2 Bayesian classifier

In this paper, we use the Bayes classifier to classify the text content of web pages. In the classifying process, the Bayes classifier outputs probabilities that a web page belongs to the corresponding categories. These probabilities also can be regarded as the similarities or dissimilarities that given web pages have with the protected web pageLet $G = \{g_1, g_2, \ldots, g_j, \ldots, g_d\}$ denote the set of web page categories, where d is the total number of categories. In fact, for anti-phishing problem only two categories are included: the phishing web page category g_1 and the normal web page category g_2.
Given a variable vector $(v_1, v_2, \ldots, v_n)$ of a web page, the classifier is employed to determine the probability $P(g_j|v_1, v_2, \ldots, v_n)$ that the web page belongs to category g_j .Applying the Bayes rule, the posterior probability $P(g_j|v_1, v_2, \ldots, v_n)$ is calculated by

$$P(g_j|v_1, v_2, \ldots, v_n) = \frac{P(v_1, v_2, \ldots, v_n|g_j)P(g_j)}{P(v_1, v_2, \ldots, v_n)}$$

where the prior probability $P(g_j)$ is estimated by the frequency of the training samples belonging to category g_j .
Naive Bayesian theory assumes that all the components in the histogram vector are independent from one another. Thus the conditional probability is represented by

$$P(v_1, v_2, \ldots, v_n|g_j) = \prod_{i=1}^{n} P(v_i|g_j).$$

The joint probability $P(v_1, v_2, \ldots, v_n)$ is described by

$$P(v_1, v_2, \ldots, v_n) = \sum_{j=1}^{d} P(v_1, v_2, \ldots, v_n|g_j).$$

Then the posterior probability $P(g_j|v_1, v_2, \ldots, v_n)$ is transformed into

$$P(g_j|v_1, v_2, \ldots, v_n) = \frac{P(g_j)\prod_{i=1}^{n} P(v_i|g_j)}{\sum_{j=1}^{c}\prod_{i=1}^{n} P(v_i|g_j)}.$$

5. Image classifier

5.1. Preprocessing and feature representation

First, we retrieve the suspected web pages and protected web pages from the web. Second, we generate their signatures, which are used for the calculation of the EMD between them. The images with the original sizes are processed into images with normalized sizes (e.g., 100×100).A signature of an image, i.e., a feature vector, is used to represent the image. It consists of features and their corresponding weights. A feature includes two components : a degraded color and the centroid of its position distribution in the image. Let F_σ = {σ,C_σ } be the feature, where σ represents the degraded color (i.e., a 4-tuple < *A*, *R*, *G*, *B* >, in which the components represent alpha, red, green, and blue, respectively), and C_σ represents the centroid of the degraded color.

The calculation of the centroid is given by

$$C_\sigma = \sum_{i=1}^{N_\sigma} (c_{\sigma,i}/N_\sigma)$$

where $c_{\sigma,i}$ is the coordinate of the *i* th pixel that has the degraded color σ, and N_σ is the total number of pixels that have the degraded color σ (i.e., the frequency). The weight corresponding to the feature F_σ is the color's frequency N_σ . Thus, a complete signature *S* is described as

$S = \{(F_{\sigma 1}, N_{\sigma 1}), (F_{\sigma 2}, N_{\sigma 2}), \ldots, (F_{\sigma N}, N_{\sigma N})\}$

where *N* is the total number of selected degraded colors.

In this signature representation, the feature weighted units in *S* are ranked in the descending order of their weights,

i.e., $N_{\sigma i} \geq N_{\sigma i+1}$ for $1 \leq i \leq N - 1$.

5.2 Distance measurement

The EMD [12], [4] is adopted to measure the distance (or dissimilarity) of two web page images, because it supports many-to-many matching for feature distributions. Suppose we have two web page images *a* and *b* with signature S_a and S_b, respectively, where S_a has *m* feature units and S_b has *n* feature units. We first calculate the distance matrix

$D = [d_{ij}]$ $(1 \leq i \leq m, 1 \leq j \leq n)$,

where $d_{ij} = D_{norm}(F_{\sigma i}, F_{\sigma j})$.

$D_{norm}(F_{\sigma i}, F_{\sigma j})$ is a normalized feature distance between feature $F_{\sigma i}$ and feature $F_{\sigma j}$, which is defined by $D_{norm}(F_{\sigma i}, F_{\sigma j}) = \mu \cdot ||\sigma_i - \sigma_j|| + \eta \cdot ||C_{\sigma i} - C_{\sigma j}||$

where $\mu + \eta = 1$. Then the flow matrix $F_{ab} = [f_{ij}]$ is calculated through linear programming and the EMD between S_a and S_b is calculated by

$$. EMD(S_a, S_b, D) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} \cdot d_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}}$$

We define the EMD-based visual similarity of two images as

$$S_{visual}(S_a, S_b) = 1 - (EMD(S_a, S_b, D))^{\alpha}$$

where $\alpha \in (0,+\infty)$ is the amplifier of visual similarity. If $S_{visual}(S_a, S_b) = 1$, the two images are completely identical, and if $S_{visual}(S_a, S_b) = 0$, the two images are completely different, because $EMD(S_a, S_b, D) \in [0, 1]$

6. Bayesian threshold estimation

We use a threshold θ in either the text classifier or the image classifier to classify a web page to be a phishing web page or a normal one. One important issue is how to appropriately set this threshold such that the number of misclassified web pages can be minimized. Anti-phishing context includes two types of misclassifications

1) false alarm: the similarity *S* is larger than θ but, in fact, the web page is not a phishing web page (false positive);

2) false negative: the similarity *S* is smaller than or equal to θ but, in fact, the web page is a phishing one.

Here, the similarity *S* is the probability $P(g_1|T)$ of the web page *T* belonging to the phishing category g_1 in the text classifier or the visual similarity S_{visual} in the image classifier In this paper, we use a Bayesian approach to model the posterior probability of a phishing web page conditioning on a specified threshold, which is proved to equally minimize the number of misclassified web pages.

Let binary state random variable $E \in \{O, N\}$ be the event that a web page is a phishing or normal one and $s \in [0, 1]$ be the similarity variable. the desired

Bayesian model to determine a posterior probability of a web page that is a phishing one conditioning on a threshold θ is given by

$$P(O|s > \theta) = \frac{P(O)P(s > \theta|O)}{P(O)P(s > \theta|O) + P(N)P(s > \theta|N)}$$

7. Fusion algorithms

One important question is how to fuse the classification results of different classifiers in a principled manner.

7.1 Weighting approach

Based on collections of similarity measurements from both text classifier and image classifier, it is straightforward to use a weight to combine the similarities into a similarity measurement as a whole. Let $Si,_T$ denote the probability that the i th web page belongs to the phishing category associated with the text classifier, and $Si,_V$ denote the similarity of the ith web page and the protected web page. The hybrid similarity measurement is defined by

$Si,_W = \beta \cdot Si,T + (1 - \beta) \cdot Si,_V$

where $\beta \in [0, 1]$ is a weighting parameter that is used to balance the weights of similarity measurements from text and image classifier. We then compare the hybrid similarity measurement Si,w to a predefined threshold θ_W, which also can be statistically estimated by using our Bayesian model. If the similarity measurement Si,W exceeds the threshold θW, the web page is classified as phishing, otherwise, the web page is classified as normal

8. Conclusion and future work

A new content-based anti-phishing system has been thoroughly developed. In this paper, we presented a new framework to solve the anti-phishing problem. The new features of this framework can be represented by a text classifier, an image classifier, and a fusion algorithm. Based on the textual content, the text classifier is able to classify a given web page into corresponding categories as phishing or normal. This text classifier was modeled by naive Bayes rule. Based on the visual content, the image classifier, which relies on EMD, is able to calculate the visual similarity between the given web page and the protected web page efficiently [4]. The matching threshold used in both text classifier and image classifier is effectively estimated by using a probabilistic model derived from the Bayesian theory. A novel data fusion model was developed . our proposed model is capable of improving the accuracy of phishing detection. More importantly, it is worth noting that our content-based model can be easily embedded into current industrial anti-phishing systems. Despite the promising results presented in this paper, our future work will include adding more features into the content representations into our current model, and knowledge updating problem in current probabilistic model.

9. References

[1] M. Wu, R. C. Miller, and S. L. Garfinkel, “Do security toolbars actually prevent phishing attacks?” in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, Montreal, QC, Canada, Apr. 2006, pp. 601–610.

[2] Y. Zhang, S. Egelman, L. Cranor, and J. Hong, “Phinding phish: Evaluating anti-phishing tools,” in *Proc. 14th Annu. Netw. Distribut. Syst. Secur. Symp.*, San Diego, CA, Feb. 2007, pp. 1–16.

[3] Y. Zhang, J. Hong, and L. Cranor, “CANTINA: A content-based approach to detecting phishing web sites,” in *Proc. 16th Int. Conf. World Wide Web*, Banff, AB, Canada, May 2007, pp. 639–648.

[4] A. Y. Fu, W. Liu, and X. Deng, “Detecting phishing web pages with visual similarity assessment based on earth mover’s distance (EMD),” *IEEE Trans. Depend. Secure Comput.*, vol. 3, no. 4, pp. 301–311, Oct.– Dec. 2006.

[5] W. Liu, X. Deng, G. Huang, and A. Y. Fu, “An antiphishing strategy based on visual similarity assessment,” *IEEE Internet Comput.*, vol. 10, no. 2, Mar.–Apr. 2006.

[6] W. Liu, G. Huang, X. Liu, M. Zhang, and X. Deng, “Detection of phishing web pages based on visual similarity,” in *Proc. 14th Int. Conf. World Wide Web*, Chiba, Japan, May 2005, pp. 1060–1061.

[7] W. Liu, G. Huang, X. Liu, M. Zhang, and X. Deng, “Phishing web page detection,” in *Proc. 8th Int. Conf. Documents Anal. Recognit.*, Korea, Aug. 2005, pp. 560–564.

[8] I. Fette, N. Sadeh, and A. Tomasic, “Learning to detect phishing emails,” in *Proc. 16th Int. Conf. World Wide Web*, Banff, AB, Canada, May 2007, pp. 649–656.

[9] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, “A comparison of machine learning techniques for phishing detection,” in *Proc. Anti-Phish. Work. Groups 2nd Annu. eCrime Res. Summit*, Pittsburgh, PA, Oct. 2007, pp. 60–69.

[10] R. Basnet, S. Mukkamala, and A. H. Sung, “Detection of phishing attacks: A machine learning approach,” in *Soft Computing Applications in Industry*, P. Bhanu, Eds. Berlin, Germany: Springer-Verlag, 2008.

[11] S. Brin and L. Page, “The anatomy of a large-scale hypertexual web search engine,” in *Proc. 7th Int. Conf. World Wide Web*, QLD, Australia, Apr. 1998, pp. 107–117.

[12] Y. Rubner, C. Tomasi, and L. J. Guibas, “The earth mover’s distance as a metric for image retrieval,” *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, 2000.

[13] Haijun Zhang, Gang Liu, Tommy W. S. Chow, *Senior Member, IEEE*, and Wenyin Liu, *Senior Member, IEEE,”,* Textual and Visual Content-Based Anti-Phishing: A Bayesian Approach” , IEEE Transactions on Neural Networks, vol. 22, no. 10, October 2011

[14] M. F. Porter, “An algorithm for suffix stripping,” *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[15] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, “On combining classifiers,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998