A Model of Detecting Phishing Websites Based on PHA and Web Noise

Jing-shi Cui, Zi-jian Wang, Bai-ling Wang*, Wei Wang and Guo-dong Xin

Department of Computing Science and Technology, Harbin Institute of Technology, Harbin, China
[a] 141110403@stu.hit.edu.cn, [b] 141110322@stu.hit.edu.cn, [c] wbl@hit.edu.cn, [d] wwhit@hit.edu.cn [e] gdxin@hit.edu.cn
* Bai-ling Wang, Wei Wang, Guo-dong Xin

Keywords: Phishing Detection, PHA, Web Noise, N-Gram.

Abstract. With the growing development of the Internet, phishing sites have caused more and more serious harm to people. Perhaps even worse, resulting in irretrievable economic losses to users. Based on the existing research, we propose a model of combining PHA and page noise to solve these problems. We use PHA to compare the similarity of those web pages' images. If we can't determine whether the site is a phishing site according to the similarity, we would extract the page noise and use n-gram to represent the site in the form of model features. After that, we compare the features of the site with the corresponding features of web pages in the corpus to determine whether it is a phishing site. Through the comparison of the experiment results with other researchers' work, we figure out that our model is simpler and more effective in the detection efficiency and other aspects.

1. Introduction

Phishing is a form of identity theft, in which criminals build replicas of target websites and lure unsuspecting victims to disclose their sensitive information such as passwords, personal identification numbers and so on. The frequent occurrence of phishing websites has seriously affected the development of online financial services and e-commerce. Also, it endangers the public interest and confidence when they use the Internet.

In order to improve the success rate of fraud, phishing web pages are generally produced by imitating the legal pages, so the pages and the target pages should exist strong relevance. This article is based on this feature and puts forward a technology of mixing PHA and page noise together [1].

According to the current research results, techniques to identify phishing sites can mainly divided into four categories. They are URL-based blacklist filtering technology, URL-based machine learning detection technology, web content-based detection technology and IP address detection technology [2]. Based on the relevant theoretical knowledge of the existing research, this paper proposes a detection model that combines blacklist filtering and web content detection.

2. Detection of Phishing Websites

2.1. Algorithm Model

2.1.1. Perceived Hash Algorithm(PHA)

PHA is a general term for comparing hashing. The main purpose is to compare and search images with similar images. PHA generates a feature string information for each picture, and then use the information to compare different images. More similar is the information, more similar are the pictures. The process of sensing the hash operation is as follows [3].

Step 1 Reduce the size of the image

First, decrease the image size to 8 pixels by 8 pixels, which means the image information is reduced to 64 pixels. Delete the high frequency and detail part of the picture, and only the structure as well as light and other basic information of the picture are left. In this way, the difference in size and proportion of the pictures can be avoided.

Step 2 Simplify the colours of the picture

Transform the 64 pixels' colour into 64 greyscales, that is, all pixels have a maximum of 64 colours.

Step 3 Calculate the grayscale average

Calculate the average of grayscales for all 64 pixels.

Step 4 Compare the grayscale of pixels

Compare the grey values of 64 pixels with the calculated grey scale average one by one. Record 0 for pixels less than the grey average and record 1 for pixels greater or equal to average.

Step 5 Calculate the hash value

Combine the above comparison results in a certain order to form a 64-bit integer. This 64-bit integer is the characteristic string information for the picture. How these 64-bit integers are combined (such as from left to right or from right to left, from top to down or from bottom to up) is not important. We just need to guarantee that all the pictures are applied to the same combination way.

After getting the feature string, we can use it to compare different pictures. We can determine how many bits of 64-bit integer are not the same by comparison, If the number of different bits does not exceed 5, then the two pictures are very similar. If the number is greater than 10, then the two pictures are not the same.

2.1.2. Hamming Distance

The Hamming distance between the two equal-length strings is the number of different characters in corresponding position of the string. Here in our model, the Hamming distance is the number of different points in corresponding position of the two binarized image points [4]. The calculation formula is as follows:

$$\mathrm{D}(\mathrm{x},\mathrm{y}) = \sum_{k=1}^{k=n} x_k \oplus y_k \qquad (1)$$

Among them, $x_k \in \{0,1\}, y_k \in \{0,1\}$.

2.1.3. Web Page Noise Extraction[5]

In addition to contents of page subject, web pages often include contents having nothing to do with the subject, such as navigation area, hyperlinks, advertising information, copyright information and so on. we define the irrelevant information as web noise. HTML source files are made up by a variety of labels and the composition content of the labels. Through the analysis of the labels, people have found that the corresponding content of some labels is full of page noise [6]. Because most of the phishing sites are imitated from the official websites, so the similarity of page noise between them is huge. Also, the noise content of the page is small and stable. If we use word features and select the page noise as a feature of the page, then we will save a lot of storage space and make the final test results more accurate. As a result, we determine whether it is a phishing site by extracting the page noises and comparing them with legal web pages [7].

By turning and referring to the relevant papers, we know that the noise in the web page source code can be divided into two categories. We get the label classification table [6]. The class 1 is almost entirely the contents of the noise, which generally account for a large proportion. The class 2 label contains almost all the content related to the web site, in which the noise content can be ignored. The classification is shown in the following table.

Table 1 Label Classification Table.

Class 1 Label	<a>,<script>,<noscript>,<style>,<meta>,<! -- -->, <param>,<button>,<select>,<optgroup>,<option>, <label>,<textarea>,<fieldset>,<legend>,<input>,< image>,<map>,<area>,<form>,<iframe>,<embed> ,<object>
Class 2 Label	<head>,<table>,,<body>,<tbody>,<p>,<Hr> , ,<tr>,<td>,<dt>,<dd>,<dl>, , ,

The noise feature extraction process is as follows: First, build a DOM tree with static HTML code and get the root [8]. Then search the tree in order to mark all the specified tags and their contents in red. Finally, extract and save all the content in red colour as a page noise feature.

2.1.4. N-Gram Language Model

The N-Gram model is a language model commonly used in continuous language recognition. When we need to convert consecutive spelling, stroke, or numbers representing letters or strokes into Chinese characters (i.e. sentences), the N-Gram model can use the collocation information between adjacent words in the context to calculate the maximum probability sentences. As a result, users do not need to select manually. Also, it avoids the problem of a lot of Chinese characters corresponding to a same pinyin (or stroke string, number string) of the re-code.

The model is based on the assumption that the occurrence of the nth word is only related to the preceding n-1 words and is not related to any other words. The probability of the whole sentence is the result of the multiplication of the probability of the occurrence of each word. These probabilities can be obtained by counting the number of simultaneous occurrences of the whole n words directly from the corpus. We commonly used binary Bi-Gram and ternary Tri-Gram [9].

If the appearance of a word depends only on the word that appears in front of it, we call it Bi-Gram. That is to say:

$$\begin{aligned} &\mathrm{P(T)} = \mathrm{P}(W_1W_2W_3 \ldots\ldots W_n) \\ &= \mathrm{P}(W_1)\mathrm{P}(W_2|W_1)\mathrm{P}(W_3|W_1W_2) \ldots\ldots \mathrm{P}(W_n|W_1W_2 \ldots\ldots W_{n-1}) \\ &\approx \mathrm{P}(W_1)\mathrm{P}(W_2|W_1)\mathrm{P}(W_3|W_2) \ldots\ldots \mathrm{P}(W_n|W_{n-1}) \end{aligned} \quad (2)$$

If the appearance of a word depends only on the two words that appear before it, then we call it Tri-Gram. In practice, we use Bi-Gram and Tri-Gram most of the time and the effect is excellent. We seldom use the model for more than four dimensions because it requires a larger training corpus and high time complexity but with little improvement in precision degree.

2.1.5. Cosine Similarity Algorithm

The cosine distance, also called the cosine similarity, is used as a measure standard of the magnitude of the difference between the two individual texts. We use Bi-Gram to convert the web pages to vectors. After this, we use cosine of the angle between the two vectors in the vector space to illustrate the difference. The closer the cosine value is to 1, the closer the angle is to 0 degree, which means the more similar the two vectors are. This is the so-called "cosine similarity".

In order to calculate the cosine between two web pages, we use x to represent the characteristic frequency vector of the protected website, and the ith characteristic frequency vector of the website under test is y_i. Finally, we employ f_{xt} and f_{y_it} to represent the tth element of the characteristic frequency vector of the protected site and the ith test site respectively. The similarity between the test sites and the protected sites is determined by calculating the angle cosine between the two characteristic frequency vectors [5]. The formula is as follows:

$$\cos(x_i, y_i) = \frac{\vec{x} \cdot \vec{y_i}}{||\vec{x}|| \cdot ||\vec{y_i}||} = \frac{\sum_{t=1}^{m+1} f_{xt} \cdot f_{y_it}}{\sqrt{\sum_{i=1}^{m+1} {f_{xt}}^2} \cdot \sqrt{\sum_{i=1}^{m+1} {f_{y_it}}^2}} \quad (3)$$

2.2. Specific Testing Process

Step 1 Check whether the site exists in the whitelist or blacklist. If it exists in the whitelist, it is determined as a legitimate website and the user can access securely. If the site exists in the blacklist, then the site is identified as a fishing web site. If the site is neither in the whitelist nor in the blacklist, we will go to the second step.

Step 2 Use PHA to process pictures. Extract the page information and calculate its hash value. If the value is less than or equal to the low threshold (set the low threshold of 5), it is considered that the site is very similar to the legitimate website and we assert it to be a phishing web site. If the value is greater than or equal to the high threshold (set the high threshold of 10), we think that the site is safe and can be accessed. Then, we add it to the whitelist. The remaining part of the value, which is between 5 to 10, we cannot determine it exactly, so we need to turn to step 3 for further judgement.

Step 3 Extract web site noise and process the collected whitelist and blacklist data by using Bi-Gram model. Then calculate the similarity values between the whitelist website and the blacklist website, and determine the threshold range of the phishing site by experimental data and graph. We can use the threshold range to judge whether a web site is a phishing site. Figure out the similarity values between the site to be tested and our whitelist sites one by one. If all of the similarity values are not in the threshold range of legal web pages, we can estimate it to be a secure site and add it to the whitelist. Otherwise as long as one of the similarity values is in the range, then the site under test is identified as phishing site and we can add it to blacklist.

2.3. Experiment and Results

2.3.1. The Collection of Data and Judgment of Accuracy

The experimental data collect from http://www.phishta nk.com/index.php. We choose six websites for our data. They are the websites of Adobe, Aol, Facebook, Google, Apple and Paypal. The official legal websites corresponding to the six sites are chosen as a whitelist. In the Phish Search column of the phishtank website, we select the phishing sites corresponding to the six sites, and then filter them in order to obtain the accessible data. A portion of the filtered URLs is used as a training set and trained in subsequent experiments to get the threshold. Another part of the URLs and some irrelevant URLs are chosen as test set to measure the algorithm.

In order to test the reliability and correctness of the experimental results, the test results were evaluated by the accuracy of the reaction algorithm called Precision, the recall of the response algorithm called Recall as well as the evaluation index of the precision rate and recall rate called F. The calculation formula is as follows [5].

$$\text{Precision}=\frac{\text{JRPW}}{\text{APW}} \quad (4)$$

$$\text{Recall}=\frac{\text{AJRW}}{\text{AW}} \quad (5)$$

$$\text{F}=\frac{2Precision\cdot Recall}{Precision+Recall} \quad (6)$$

In order to verify the rationality of our algorithm more intuitively and accurately, our experimental part mainly focuses on the detection of PHA in the second step and the detection of the page noise in the third step.

2.3.2. PHA Detection

Use the PHA to calculate the hash of the sites in whitelist and the sites to be measured. Then calculate the Hamming distance between the hash sequences. Sites with Hamming distances less than 5 are identified as phishing sites and more than 10 are identified as legitimate sites.

2.3.3. Web Noise Detection

Use equation (3) to calculate the similarity values of legal web sites in whitelist and the phishing sites in blacklist, get the following distribution figure.

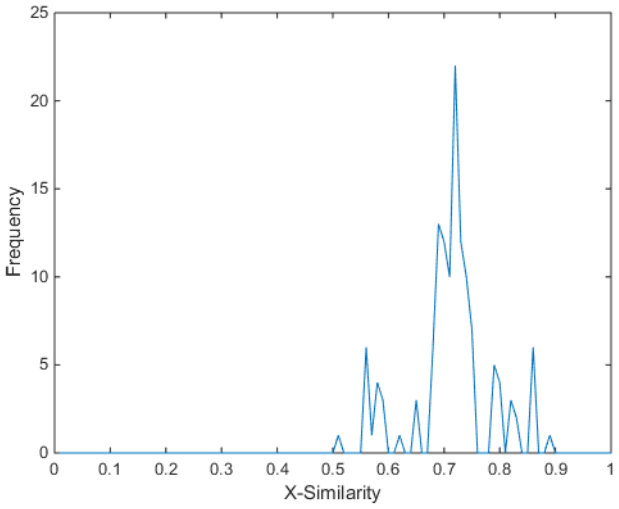

Figure 1 Similarity distribution of whitelist sites and corresponding training set.

As can be seen from the above figure, similarity values are concentrated in about 0.72, through which we can roughly identify the troubleshooting threshold range of phishing sites is between 0.69 to 0.75.

2.3.4. Comparison

Compare the results of the two steps in the detecting process, we get the results shown in the following table.

Table 2 The Accuracy of Each Step of the Experiment.

	Precision	Recall	F
PHA Detection	71.43%	70.97%	71.20%
Web Noise Detection	88.26%	85.26%	86.73%

Through the above table, we can figure out that with the gradual progress of the two steps, Precision value and Recall value gradually increase, indicating that the algorithm is more accurate and has a higher coverage rate.

At the same time, the Precision value of this experiment is higher than the current detection algorithm. Hence, Precision value has been improved.

3. Conclusions

At present, there are many ways to detect phishing sites, these methods have many aspects worthy of our reference and learning, but also there are some limitations. Therefore, based on the previous research, we put forward some new detection ideas.

First, determine whether the test site has been found in the blacklist or white list. This is the most basic step which has been found in most detection algorithm. Then, use PHA to compare the image similarity. Because most of the phishing sites are made of genuine legal sites, so in order to make users deceived, most phishing sites are similar to legitimate pages. Through the comparison of picture similarity, we can determine some of the phishing sites and legitimate sites. Finally, because some test sites still cannot be judged with the method of PHA, so we need further judgments. At this point, we select the web page noises with few changes as features to describe the entire page. Then, use n-gram algorithm to deal with the noises and describe them in the form of probability

vector. By comparing the similarity of the feature matrices, the threshold range of phishing sites is obtained. Finally, a complete algorithm for detecting phishing sites is established. At the same time, the rationality of the algorithm is verified by experiments.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 61371177 and No. 61170262), National Key Research and Development Plan under grant 2016YFB0800802, Science and Technology Major Project in ShanDong under grant 2015ZDXX0201B04, Science and technology development Program in Shandong Province under grant 2014GGX101053, Science and technology development Program in Shandong Province under grant 2014GGX101053, Science and technology development Program in Weihai Province under grant 2014GGX101053. We would like to thank the anonymous reviewers for their helpful comments.

References

[1] GUANG XIANG, JASON HONG, CAROLYN P. ROSE, LORRIE CRANOR. CANTINA+: A Feature-rich Machine Learning Framework for Detecting Phishing Web Sites[J]. ACM Journal Name, Vol. V, No. N, Month 20YY, Pages 1–32

[2] Prakash P., Kumar M.. Kompella R. R., et al. PhishNet: Predictive Blacklisting to Detect Phishing Attacks[C]. In: Proc. of IEEE INFOCOM, IN, 2010: 1-5

[3] LU Kang, ZHOU An-min. Phishing-Website Detection based on Image Similarity[J]. Technology & Research, 2016, Pages 116

[4] YANG Jian-cheng, HOU Zhang-yong, ZHANG Xiao-jin. Phishing site detection and protection based on SDN[J]. JOURNAL OF MINJIANG UNIVERSITY, 2015, NO. 148, Pages 69-75

[5] YIN Lan-fang. The Research on Phishing Detection using Webpage Noise and n-gram[D]. Changsha: Central South University of Forestry and Technology, 2015

[6] Zou Yong-qiang, Zhong Zhi-nong. An Efficient Approach to Reduce Noise in News Webpages [J]. MICROCOMPUTER ITS APPLICATIONS, 2011, Vol 30, No. 16, Pages 64-71

[7] Debnath S., Mitra P, Pal N., et al. Automatic identification of informative sections of Web pages [J]. IEEE Trans on Know ledge and Data Engineering, 2005, 2(17): 1233-1246

[8] Lin S. H., Ho J. M.. Discovering informative content blocks from Web documents. [C]. In: Proc of the 8th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2002: 588- 593

[9] YU Yi-Jiao, LIU Qin. N-gram Chinese Characters Counting for Huge Text Corpora[J]. Computer Science, 2014, Vol 41, No. 4, Pages 263-268