Journal of Computational Information Systems 9: 14 (2013) 5553–5560
Available at http://www.Jofcis.com

An Efficient Approach to Detecting Phishing Web ⋆

Xiaoqing GU, Hongyuan WANG*, Tongguang NI

School of Information Science and Engineering, Changzhou University, Changzhou 213064, China

Abstract

As the Electronic Commerce and On-line Trade expand, phishing has already become one of the several forms of network crimes. This paper presents an automatic approach for intelligent phishing web detection based on learning from a large number of legitimate and phishing webs. As given a web, its Uniform Resource Locator(URL) features are first analyzed, and then classified by Naïve Bayesian(NB) classifier. When the web's legality is still suspicious, its webpage is parsed into a document object model tree, and then classified by Support Vector Machine(SVM) classifier. Experimental results show that our approach can achieve the high detection accuracy, the lower detection time and performance with a small sample of the classification model training set.

Keywords: Phishing; Naïve Bays(NB); Support Vector Machine(SVM); Classifier

1 Introduction

Phishing is a major security threat to the online community. It is a kind of identity theft that makes use of social engineering skills and technical subterfuge to entice the unsuspecting online consumer to give away their personal information and financial credentials [1]. A typical phishing attack consists of four phases, namely, preparation, mass broadcast, mature, and account hijack [2]. In order to direct users to fraudulent webs and steal their money, phishing patterns evolve constantly by phishers. Generally, most phishing webs use links pointed to legitimate webs and visually similar content to lure visitors to enter their sensitive information. In this sense, phishing webs are not isolated from their targets but have strong relationships with them, which can be used as clues to find their targets.

To detect and prevent various kinds of phishing attacks, there are many different preventive strategies and detective ideasInformation security specialists and anti-phishing organizations have set up phishing alerts databases that assess each reported phishing incident in terms of its risk level. The blacklist-based anti-phishing toolbars are developed by many companies such as Netcraft [3], Google Toolbar [4]. Ying Pan et al. [5] have invented a phishing website detection system which examines the anomalies in web pages, it demands neither user expertise nor prior knowledge

⋆Project supported by the National Nature Science Foundation of China (No.61070121).
*Corresponding author.
Email address: hbxtntg-12@163.com (Hongyuan WANG).

1553–9105 / Copyright © 2013 Binary Information Press
DOI: 10.12733/jcis6350
July 15, 2013

of the website. Mingxing He et al. [6] have invented an efficient phishing webpage detector. It converts a webpage into 12 features and determines whether a webpage is a legitimate or a phishing one using an SVM classifier. Xi Chen et al. [7] have adopted a hybrid text and data mining model that used key phrase extraction technique to discover important semantic categories from the textual content of the phishing alerts, and to come up with classification of risk level of the attack and the loss in market value of the firm. Huaibin Wang et al. [8] have invented an method by checking phishing pages using vector features which determine the fundamental structure of a web page and these features are used to measure the similarity by similarity algorithm. Weiwei Zhuang et al. [9] have proposed an intelligent anti-phishing framework using multiple classifiers combination, and introduced the incremental bagging method to improved feature selection algorithm cuts the "redundant" features.

In this paper, a fast and accurate approach is proposed to detect phishing web. Our approach determines whether a webpage is a phishing web or a legitimate one, based on its URL and webpage features, and is merely a combination of NB and SVM. The NB classifier used to detect the URL is that NB is a rapid detection method for classification and URL features can be easily acquired. If the NB classifier cannot judge the given web's legality definitely, the SVM classifier is used to detect it based on its webpage features. Also our approach may work together with a blacklist-based method to provide a better protection, as the CANTINA [10].

Section 2 describes the approach of our work. Section 3 describes the URL features used for detecting the phishing and NB classifier. Section 4 describes the webpage features used for detecting the phishing and SVM classifier. Experimental results are presented in Section 5. Finally the conclusion is given in Section 6 and point out the future work.

2 Approach

2.1 System architecture

In this section, a detailed discussion of our approach is provided to classifying webs reputation. The system architecture is shown in Fig. 1. Our approach is performed in the following proce-

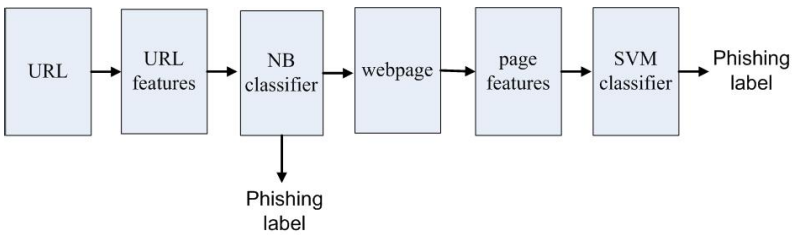

Fig. 1: System architecture

dures:

Step 1 Given a web P, extract its URL identity and generate features.

Step 2 Classify P by NB classifier and return result (+1, -1 or 0).

//+1: legitimate, -1: phishing, 0: suspicious

Step 3 If result=+1 or -1, output the phishing label,
If result=0, go to Step 4.

Step 4 If P has not a text input, output the phishing label (1).
If P has a text input, go to Step 5.

Step 5 Extract its webpage identity and generate features.

Step 6 Classify P by SVM classifier and output the phishing label.

2.2 Identity extraction

Classing a URL with a trained model is a lightweight operation compared to first downloading the webpage and using its content for classification. For our purposes, URL reputation is treated as a binary classification problem where legitimate examples are benign URLs and phishing examples are malicious URLs. Significantly, webs are classified based only on the context of the URL and the relationship between URLs and the lexical. The features of URL are consulted by the studies of McGrath, Gupta [11] and Justin Ma et al. [12].

Identity of a webpage is a set of words that uniquely identifies the ownership of the website. The webpage identity is retrieved from two sources; one is from the content of a webpage and the other is from the structure of a webpage. Therefore these features are useful to find the identity of the web page. Features extracted in identity extraction phase include META Title, META Description, META Keyword and HREF of $< a >$ tag.

Here the webpage is parsed into the Document object model (DOM) tree. DOM [13] is a platform and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. Similar to [6, 14] approach, the term frequency-inverse document frequency (tf-idf) technique is applied to extract term identity set from a webpage.

META Tag: The $< meta >$ tag provides metadata about the HTML document. Meta elements are typically used to specify page description, keywords, author of the document, last modified and other metadata. The Meta description tag is a snippet of HTML code that comes in the Head section of a web page. It will be placed before the Meta keywords tag. The identity relevant object is the value of the content attribute in Meta tag. It consists of a description about the web.

HREF Tag: The href attribute specifies the destination of a link. When a hyperlink text is selected, it has to direct to the concerned web page. Phishers will not perform any change in the destination site address. So it points to the legitimate web. The value of the href attribute is a URL in which the domain name has high probability to be the identity of the web.

After tokens or terms are produced in the tokenization process, the tf-idf value of each token is counted. tf-idf weight is evaluated for each of the keywords, and tf-idf value is calculated using the following formula:

$$tf_{i,j} = \sqrt{\frac{n_{i,j}}{\sum_k n_{k,j}}} \tag{1}$$

where $n_{i,j}$ is the number of occurrences of term t_i in document d_j, and the denominator is the number of occurrences of all terms in document d_j.

The inverse document frequency measures the importance of a term in a collection of documents. The inverse document frequency idf_i of term t_i is defined as:

$$idf_i = ln(\frac{|D|}{|\{d_j : t_i \in d_j\} + 1|}) + 1 \tag{2}$$

where $|D|$ is the total number of documents in a dataset, and $|\{d_j : t_i \in d_j\} + 1|$ is document frequency. To find the document frequency of a term, WebAsCorpus is used. It is a readymade frequency list. The total number of documents in which the term appears is the term that has the highest frequency. The tf-idf weight of term t_i in document d_j is simply a multiplication of the tf and idf value:

$$tf - idf_{i,j} = tf_{i,j} \cdot idf_i \tag{3}$$

Finally terms are retrieved whose tf-idf values are ranked top five to be the term identity set of a webpage.

3 URL Features and NB Classifier

3.1 URL features

Feature extraction plays an eminent role for the efficient prediction of phishing web. According to section 2, there are four features including. The features are described as the following:

IP Address: For escaping from domain registration or user checking, the IP address is a simple way used to hinder from verification.

Dots in URL: Many dots appearance may be caused by an attempt that the phishing web use sub-domain to construct a legitimate look of the URL or use a redirect script to bring the victim to another site. Here the number of dots in a page's URL is checked.

Suspicious URL: When the phishing web try to trick the victims, the URLs of the phishing web may be modified to the pattern that is hard to check. '@' or '-' signs in suspicious URLs is checked which are often used to modify the URL.

Slash in URL: The URL should not contain more number of slashes. If it contains more than five slashes then the URL is considered to be a phishing URL .

3.2 NB classifier

The features described are used to encode webs' URLs as high dimensional feature vectors. The N-B classifier is considered one of the most effective approaches for learning how to classify text documents [15]. Given a set of classified training samples, an application can learn from these samples so as to predict the class of an unmet sample. Each URL is represented by features$(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$ are independent from each other. Each feature $\mathbf{x}_i (1 \leq i \leq 4)$ takes a binary value(0 or 1) indicating whether the corresponding property appears in the URL. The probability is calculated that

the given web belongs to a class c(c1:legitimate and c2:phishing) as follows:

$$p(C_i|\mathbf{X}) = \frac{p(C_1) \times p(\mathbf{X}|C_i)}{p(\mathbf{X})} = \frac{p(C_1) \times \prod_{i=1}^{4} p(\mathbf{x}_i|C_1)}{p(\mathbf{X})} \tag{4}$$

where all of $p(\mathbf{X})$ are constant, meanwhile $P(\mathbf{x}_i|c_1)$ and $P(C_i)$ can be calculated easily from training.The proportional to $\frac{P(C_1|\mathbf{X})}{P(C_2|\mathbf{X})}$ is calculated, and the results are as follows:

$$\begin{cases} \frac{P(C_1|\mathbf{X})}{P(C_2|\mathbf{X})} > \alpha \ \ (\alpha > 1), & a\ legitimate\ web, \\ \frac{P(C_2|\mathbf{X})}{P(C_1|\mathbf{X})} > \alpha \ \ , & a\ phishing\ web, \\ 1/\alpha \leq \frac{P(C_1|\mathbf{X})}{P(C_2|\mathbf{X})} \leq \alpha, & a\ suspicious\ web,\ need\ to\ be\ detected\ further. \end{cases} \tag{5}$$

For the suspicious web, SVM is used to detect it according to web's content features.

4 Webpage Features and SVM Classifier

4.1 Webpage features

Given a suspicious web P and its term identity generation step would determine the features value of the webpage. The feature vector generated in this step would then be inputted into a SVM classifier to determine whether a web is a phishing or a legitimate web. The features are categorized that are gathered for web's content as follows:

Forms: If a page contains any HTML text entry forms asking for personal data from people, such as password and credit card number. The HTML is scanned for $< input >$ tags that accept text and are accompanied by labels such as "credit card" and "password". Most phishing webs contain such forms asking for personal data, otherwise the criminals risk not getting the personal information they want.

Nil anchors: A nil anchor is an anchor that points to nowhere. The more nil anchors a page has, the more suspicious it becomes.

Foreign Anchor: An anchor tag contains href attribute whose value is an URL to which the page is linked with. If the domain name in the URL is not similar to the domain in page URL then it is called as foreign anchor. For any web, it is normal to link to the foreign domains, but too many foreign anchors would decrease the credibility of the web.

Foreign requests: Similar to the foreign anchors, requests to the foreign domains are also a normal behavior. When there are too many foreign requests, the web could be less credible.

Foreign Anchor in Identity Set: A foreign anchor is an anchor that points to a foreign domain. Foreign anchors in identity set in a webpage are suspicious, since phishing pages often have the majority of its anchors pointing to the legitimate web to imitate the behavior, thus the URL identity which the phishing page claims is the legitimate web domain.

Foreign request in Identity set: To imitate the real web, phishing pages might request images, Javascript, CSS files and other objects from the real web. For each foreign request, the domain is compared with the URL identity if the URL identity is in a foreign domain; otherwise,

if the URL identity is in the local domain, then the domain of the request URL is compared to the term identity set.

SSL Certificate: SSL is an acronym of secure socket layer. It creates an encrypted connection between the web server and the user's web browser allowing for private information to be transmitted without the problems of eavesdropping. All legitimate webs will have SSL certificate. But phishing webs do not have SSL certificate.

Based on above features, the Forms feature is used to be a filter for dataset selection. The reason is that the dangerous pages causing users lost their information must contain forms with input blocks. If a webpage has not a text input, the detection is not required since users do not have a way to enter their secret information.

4.2 SVM classifier

SVM as a well-known data classification technique is applied to classify webpage features. The SVM classifier input in our approach is a 6-dimension feature vector produced from the feature generation step ($V_P =< F_1, F_2, F_3, F_4, F_5, F_6 >$). Since a webpage is only considered as a legitimate or a phishing, it is naturally a binary classification problem. The SVM would produce output in two classes: -1 means phishing, and +1 means legitimate. Here the least squares support vector machine(LS-SVM)is applied, and the optimal model is as follows

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{6}$$

Where $K(\mathbf{x}, \mathbf{x}_i)$ is the RBF kernel(SVM-rbf), and the form is $K(\mathbf{x}, \mathbf{x}_i) = e^{-\gamma}\|\mathbf{x} - \mathbf{x}_i\|^2$, $\mathbf{x}$, $\mathbf{x}_i$ are webpage features. The value of α and b can be obtained by solving the following equations:

$$\begin{vmatrix} 0 & 1 & \cdots & 1 \\ 1 & K(\mathbf{x}_1, \mathbf{x}_1) + 1/\gamma & \cdots & K(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \vdots & & \vdots \\ 1 & K(\mathbf{x}_n, \mathbf{x}_1) & \cdots & K(\mathbf{x}_n, \mathbf{x}_n) + 1/\gamma \end{vmatrix} \begin{vmatrix} b \\ \alpha_1 \\ \vdots \\ \alpha_n \end{vmatrix} = \begin{vmatrix} 0 \\ y_1 \\ \vdots \\ y_n \end{vmatrix} \tag{7}$$

Before the classifying of SVM, it should undergo a training process to develop a classification model. $\mathbf{x}_i$ and y_i $(i = 1, \ldots, n)$ indicate the Feature vectors and class label of the web samples which are identified classified. If $f(\mathbf{x}) = +1$, the giving web is considered to be a legitimate one, and if $f(\mathbf{x}) = -1$, the giving web can be considered to be a phishing one.

5 Experiments and Results

The dataset used for learning is collected from PHISHTANK [16]. The dataset with 600 phishing webs and 400 legitimate webs is developed for implementation. 100 legitimate and 100 phishing webs are taken as the training set, and the rest of 300 legitimate and 500 phishing pages compose the testing dataset. The robustness of the classifiers is evaluated using 10-fold cross validation. The feature vector corresponding to phishing web is assigned a class label -1 and +1 is assigned to legitimate web.

Two experiments have carried out to evaluation of our method. In the first experiment, the optimal value of α is searched for. For comparison of the experiments results, two evaluation metrics used commonly:

True Positive (TP)-The phishing webs that were classed as phishing web.

False Positive (FP)-The legitimate webs that were classed as phishing web.

Table 1: The value of α and performance

α	TP (%)	FP (%)	α	TP (%)	FP (%)
1.1	90.68	4.78	2.1	95.34	1.32
1.2	91.32	4.31	2.2	96.63	0.74
1.3	91.91	2.93	2.3	97.35	0.31
1.4	92.30	4.08	2.4	97.71	0.13
1.5	92.99	3.69	2.5	97.71	0.13
1.6	93.06	3.78	2.6	97.71	0.13
1.7	93.58	3.15	2.7	97.72	0.13
1.8	93.89	2.78	2.8	97.71	0.13
1.9	94.20	3.01	2.9	97.71	0.12
2.0	94.88	1.73	3.0	97.71	0.13

The performance result is shown in Table 1. The value of α is tested between 1.1 and 3.0. The TP rate of NB classifier is raising to maximum 99.73% when α is equal to 2.4 and nearly keep the same when between 2.5 and 3.0. The FP rate of NB classifier is dropping to minimum 0.13% when α is equal to 2.4. So our approach provides the best performance with α= 2.4.

In the second experiment, the accuracy of the three classifiers is compared: NB, SVM and our approach. The features describing the properties of URL and webpage are both used in NB classifier and in SVM classifier, including 11 features in all that are described in section 3 and 4.

Table 2: Performance comparison

–	NB	SVM	Our Approach
Train Time	50s	92s	71s
Test Time	80s	109s	90s
TP(%)	90.08	94.41	96.90
FP(%)	4.80	3.98	1.25

Table 2 shows the training, testing times and detection accuracy for each classifier. Based on the comparison in Table 2, the accuracy of our approach outperforms the other approach while its false alarm rate is much lower than the other approach. The training and testing time of NB is shortest, but the accuracy is lowest. The training and testing time of SVM is longest, and the accuracy is lower than our approach.

6 Conclusion

In this paper, a novel approach is presented to identifying the potential phishing target of a given web. Every web claims a webpage identity, either real or fake. If a web claims a fake identity, abnormality may exist in a network space; therefore our approach could detect and differentiate between a legitimate and a phishing web. Our approach first categorizes the URL features and test whether the page is phishing or not using NB. When the web's legality is still suspicious, then categorize its webpage features and test whether the page is phishing or not using SVM. The experimental results show that our approach has a high detection rate and a low false positive rate. In future works, the plan is to adjust existing feature extraction methods and seek for more relevant features to get a better result.

References

[1] J. S. Downs, M. B. Holbrook, Decision strategies and susceptibility to phishing, in: Proc. the second symposium on usable privacy and security(SOUPS 2006), pp. 79-90.

[2] I. Bose, A.C.M. Leung, Unveiling the mask of phishing: threats, preventive measures and responsibilities, Communications of the Association for Information Systems 19 (24) (2007) 544-566.

[3] Google Inc, Google safe browsing for Firefox, http://www.google.com/tools/firefox/ safebrowsing/

[4] Netcraft Inc, Netcraftanti-phishing toolbar, http://toolbar.netcraft.com/

[5] Y. Pan, Anomaly based web phishing page detection, in: Proc. Twentysecond annual computer security applications conference(ACSAC'06), 2006, pp. 381-392.

[6] I. He, S.J. Horng, An efficient phishing webpage detector, Expert Systems with Applications 38 (2011) 12018-12027.

[7] X. Chen, I. Bose, Assessing the severity of phishing attacks: A hybrid data mining approach, Decision Support Systems 50 (2011) 662-672.

[8] H.Wang, B.Zhu, C.WANG, A Method of Detecting Phishing Web Pages Based on Feature Vectors Matching, Journal of Information and Computational Systems 2012 Vol. 9 (15): 4229-4235.

[9] W.Zhuang, Q. Jiang, Intelligent Anti-phishing Framework Using Multiple Classifiers Combination, Journal of Computational Information Systems 2012 Vol. 8 (17): 7267-7281.

[10] Y. Zhang, J.Hong, CANTINA: A content-based approach to detecting phishing web sites, in: Proc. the international World Wide Web conference(WWW), 2007, pp. 639-648.

[11] D. K. McGrath,M. Gupta, Behind Phishing: An Examination of Phisher Modi Operandi, in: Proc. the USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), 2008, pp. 1123-1136.

[12] J. Ma, L. K.Saul, S.Savage, Identifying Suspicious URLs: An Application of Large-scale Online Learning, in: Proc. of International Conference on Machine Learning, 2009, pp. 681-688.

[13] W3C DOM Interest Group, Document object model, W3C DOM Interest Group, http://www.w3.org/ DOM/, 2012.

[14] S.L.Va, V.MSb, Efficient prediction of phishing websites using supervised learning algorithms, Procedia Engineering 30 (2012) 798-805.

[15] W.Han, Y.Cao, Using automated individual white-list to protect web digital identities, Expert Systems with Applications 39 (2012) 11861-11869.

[16] PhishtankInc, phishing dateset, http://www.phishtank.com/.