Research Journal of Information Technology 6(3): 39-43, 2015
ISSN: 2041-3106; e-ISSN: 2041-3114

Submitted: May 17, 2015 Accepted: July 10, 2015 Published: August 05, 2015

Research Article

Email Phishing Detection System Using Neural Network

Amina A. Abdullah, Loay E. George and Imad J. Mohammed
College of Science, University of Baghdad, Iraq

Abstract: One of the internet based identity thefts is called phishing. Email phishing is a way that phishers trick the user to give information. The increasing of the phish attack in the resent years cause a lot of problems; credit card number, user name, password were stolen due to the phish attack. Due to this attack people lose their money, personal information and trust in online business also the attack effects on the companies' reputation. Many companies were losing money up to millions of dollars. In this study a new stem that can quickly detect phishing emails with low false positive rate is introduced. A set of features is proposed to test each coming emails to identify whether it is phish email or not. A feed forward neural network with back propagation training algorithm was adopted to categorize the email samples into phish or ham category. First a set of extracted features vectors from each email, each vector consists of 17 features, had been used to make the categorization decision. Then, a set of smaller feature vectors, each consists of only the 12 best features, was used for classification tests. The results achieved in this study are 99.91 (for 17 features) and 99.95for (12 best features).

Keywords: Email phishing, email threats, machine learning (neural network), phishing detection

INTRODUCTION

Internet is playing an increasingly significant role in today's commerce and business activities (Chandrasekaran *et al*., 2006). Email is a very useful tool for coordination, as it enables the dissemination of information to a large number of recipients at a very low cost (Kavada, 2007). Phishing is a form of internet scam in which the attackers try to trick consumers into divulging sensitive personal information. The techniques usually involve fraudulent E-mail and web sites that impersonate both legitimate E-mail and web sites (Tally *et al.*, 2004). Phishing emails pose a serious threat to electronic commerce because they are used to defraud both individuals and financial organizations on the Internet (Almomani *et al*., 2013).

Financial institutions are at risk for large numbers of fraudulent transactions using the stolen information (Tally *et al.*, 2004). For the first time, Apple was the world's leading phishing target, with 21,951 attacks (17.7% of all attacks) Perennial targets PayPal (17,811 attacks, or 14.4%) and Taobao.com (16,418 attacks, or 13.2%) were second and third according to APWG report in the second half of 2014 (Aaron *et al*., 2014). There were at least 123,972 unique phishing attacks worldwide in the second half of 2014 this was almost exactly the same number as in the first half of 2014 (Aaron and Rasmussen, 2015). In 2010, 43% of all the OSM (Online Social Media) users were targets of phishing attacks. In 2012, around 20% of all phishing attacks targeted Facebook (Aggarwaly *et al*., 2012).

In this study a standard dataset of 9100 emails have been used to detect phishing attack. Among the properties may appear in email can be used to identify phish and ham emails, 17 features (properties) have adopted as signs for email type. The significance of each feature had been weighted to find the best features. A subset consist of 12 features was allocated as the best features vector. The feed forward neural network model was used to train and test the samples. First, the system was trained and, then, tested using feature vectors each hold the whole 17 extracted features. As next step, the system was retrained and tested using only 12 features and the final results were compared.

LITERATURE REVIEW

Ghazhie and George (2013) have proposed a new approach to detect phishing emails. They proposed two models to detect email phishing attack, in the first model they used 18 semantic features which appear in header and HTML body. Three detection methods have been adopted to make decisions to identify phish and ham emails. One of these methods is based on a new approach which is based on feature existence and feature decisive value criteria. The other two applied methods are K-Nearest neighbor algorithm and Feed Forward Neural network. The new approach suggests weighting the features; and according to this approach the authors selected 7 best features from 18 features. With k-nearest neighbor and feed forward neural network methods the taken training set was consist of 6000 emails, while for testing purpose a subset consists

Corresponding Author: Amina A. Abdullah, College of Science, University of Baghdad, Iraq

of 3100 emails was used. In their second suggested model (Ghazhie and George, 2013) they suggested using a set of 45 significant words which may appear in the text body of the email. Two categorization methods have been used; they are:

- Word existence and word decisive value criteria
- k-nearest neighbor. They used their new method to weight the word and they found best 11 words. They trained the neural network with 6000 emails and they used 3100 emails to test the model. They used two data sets, ham data set and phish data set each data set contains 4550 emails. They claimed that the best result they got using k-nearest neighbor achieved 98.92% with features best model and 91.99% with word best model.

Khonji *et al*. (2012) proposed an enhancement to LUA (Lexical URL Analysis) which was created by them in previous study (Khonji *et al*., 2011). The study claims an enhancement for anti-phishing email filtering is accomplished. The suggested method uses random forest machine learning with LUA. The method is used to detect phish website pages and phish emails. They collected a data set of website of total 25428 legitimate URLs and 48305 phish URLs from Khalifa university's HTTP proxy and volunteers whom used their experimental HTTP proxy server, the email data set is 4116 phish emails and 4150 ham emails. For website classification they used LUA tokenization mechanism. The best result they got is 97.31%. In the email classification model they used features that were extracted from email and they added the LUA as a feature, where the highest value of LUA represents as phish value. They used two sets of features; the first set is without LUA feature, it contains 47 features. While the second set contains the same features of first set with the LUA feature. They used a wrapper and best-first as the feature subset selection method, the result was 6 subsets: 3 subsets with LUA feature and 3 subsets without LUA feature. They used 90% of data to train the system with random forest and 10% to test the system while they used 30% of data to train the system with LUA and 70% to test the system. The result they got is 99.45% with low false positive rate.

Vaishnaw and Tandan (2015) proposed a hybrid system to detect phishing email. They used 47 features and 8266 emails to detect. They used many machine learning algorithms: Bayesian, CART (classification and regression technique, CHAID (Chi-Squared Automation Interaction Detection), ANN (Artificial Neural Network), SVM (support vector machine), decision tree, C5.0 and QUEST. They used bagging and boosting technique to combine the models. First they trained and tested each model alone to find the result of each model, then they used a combination of the two or more models, they depend on the accuracy result that they got to choose the best system. They combined Bayesian net model with the other models to get good results. They refereed that their best achieved result is 99.32% when the combination of CART and Bayesian is used.

Fette *et al*. (2007) proposed a new method to detect phishing emails called PILFER. They used phish data set contains 860 emails and ham data set contains 6950 emails. They used 10 features in their system. They referred the method result is 99% and their methods is better than SpamAssassin. Also, the achieved rate of false positive is 0.1%.

METHODOLOGY

Two data sets have been used in this study, phish data set and ham data set, each data set contains 4550 emails. In this study, a set 17 features that commonly appeared in phish emails have been used in the phishing

Table 1: The features used for phish email detection

Feature no.	Description
1	If text/html in the content type of email, the feature takes 1 otherwise it is 0
2	If from part of email header equal to the reply-to part this feature takes the value 1 otherwise 0
3	If <form in the html body of email this feature set to 1 otherwise it take the value 0
4	If <script> or <javascript> tag in the html code in html body the feature takes value 1 otherwise it take the value 0
5	If URL in the html body is more than 3 this feature takes value 1 otherwise 0
6	If from part in email header is not equal to any of domains in URL this feature is set to 1 otherwise 0
7	This feature is 1 if the dot in the domain of any links in the html body is more than 3 otherwise it is 0
8	If the number of pictures is used as links is more than 2 this feature is set to 1 otherwise it is 0
9	If the domain in URL is more than 3 this feature takes 1 otherwise it is set to 0
10	If the symbol '@' is in the URL this feature takes the value 1 otherwise it takes 0
11	If the ports that used as link in URL is not in list of legal port; this feature takes the value 1 otherwise it takes the value 0. List of legal ports {':80', ':80/', ':443', ':443/'}
12	If hexadecimal characters are used in URL this feature takes the value 1 otherwise 0
13	If the IP based URL is used the feature takes the value 1 otherwise 0
14	The value 1 is taken if one of the words: 'click', 'click here', 'log' and 'login' are appeared in text URL. Otherwise the feature takes the value 0.
15	This feature takes the value 1 if the text URL is not matched with the target URL otherwise the feature takes the value 0
16	The value 1 is taken if the attacker attempts to use fake secure socket layer (SSL) in the text URL to trick the user. In this case the 'https' appear in text URL while the target URL is 'http'. Otherwise the feature takes 0.
17	If the size of email is less than 25 KB the feature takes the value 1 otherwise the feature takes the value 0.

detection system; Table 1 shows the 17 features and a short description for each feature. These features are extracted from emails and the best features are founded using fist order statistical measures. Artificial feed forward neural network with back propagation algorithm is used to train and then identify phish and ham emails. The system was trained and tested twice; first using all the 17 extracted features and then using the best features. The system passed through two phases; the first phase is training using back propagation algorithm and the second phase is phish emails detection.

Preprocess phase: Email consists of header and body. The features are extracted from the email header fields ("From", "Reply-to", "Content-type") and from HTML body. The pre-processing phase consists of two main steps:

Feature extraction: 17 features have been extracted for each email. For each data set a two dimensional array consists of 4550 row (number of emails) and 17 column (number of features) was used to store the features (i.e., one vector for each email). The array of features contains only 1s' and 0s'. The value 1 was used to indicate the feature existence and the value 0 to indicate that the feature is not existed.

Select best features: The occurrence of each feature in the phish and ham emails is different. The selection of each feature depends on two basic factors: The frequency of occurrence of the feature in phish and ham emails, the difference between the phish and ham preparation. The following steps were used to find the best features:

- Calculate the number of occurrence of each feature in both ham and phish emails, the calculated occurrence frequencies of all features belong to certain phish or ham email file are saved in a vector.
- Calculate the probability (p) of occurrence of each feature in ham and phish emails, separately using the following equation:

$$p_{phish}(i) = \frac{Frequency\ of\ occurrence\ of\ the\ i^{th}\ phish\ emails}{Total\ Number\ of\ tested\ phish\ emails} \times \%100$$

$$p_{ham}(i) = \frac{Frequency\ of\ occurrence\ of\ the\ i^{th}\ ham\ emails}{Total\ Number\ of\ tested\ ham\ emails} \times \%100$$

where, $p_{phish}(i)$ is the probability of i^{th} feature in phishing emails, $p_{ham}(i)$ is the corresponding probability in ham emails; iϵ[1,17]. The values of $p_{phish}()$ and $p_{ham}()$ are stored in phish or ham vectors, respectively.

- Find the highest probability of each feature for the ham and phish emails; and they are compared. If the probability of feature occurrence in phish emails is much higher than its probability in ham emails; this means that the feature is a good discriminating feature. If the percentage in phish is equal or less than the percentage in ham feature, the feature will be registered in features discard list.
- Calculate the relative difference between ham and phish probabilities: If the feature in the previous step was not added to discard list then the difference is calculated; simply it is a subtraction of probability occurrence in ham emails from the its corresponding probability in phish emails. If the calculated relative difference is high, the feature will be considered a good feature, otherwise the feature is added to discard list.
- **Remove unwanted features:** The last step is to delete all the discarded features. The good features that remained are 12 features.
- **Select train and test vectors:** This step is done for both cases of vectors (with 17 features and with 12 best features features). A statistical analysis was conducted on the established features vectors and it found that most of the features appear more than once and their repetition numbers are different. The following steps are taken to remove the redundancy and select a reduced set of non-redundant templates of feature vectors:

  - The first feature vector in the phish array is added to the unduplicated array; then each vector in phish array is compared with all vectors in the unduplicated array, if the tested vector does not exist then the feature vector is added to the unduplicated array. Same procedure is done with ham array.
  - **Extract the common vectors:** Extract the vectors that appears in both arrays (unduplicated ham and phish), these vectors is added to the array of common vectors.
  - **Count the occurrence of the common vectors:** The occurrence of each common vector in phish array is calculated. Same steps are done for ham array.
  - If the common vector appears in ham array less than it appears in phish array, then the vector will be discarded from the ham array and unduplicated ham and vice versa. Also, if the vector appears equally in both ham and phish, this vector is removed from both ham and phish arrays and unduplicated arrays too.

After accomplishing the above steps, then the final arrays (feature array and unduplicated array of both ham and phish) have non redundant vectors.

Detection phase: This phase is consists of two phases (i.e., the training and testing phases), the steps of these phases are:

Training: The array of unduplicated vectors was used as training, which is fed as input for back propagation for training the feed forward neural network. The neural network was trained using different learning rates and momentums to find their proper values. The proper weights were found to be between (0.001-0.0001). The output of training the neural network is a set of weights that used in the testing phase.

Testing: In this phase the test set of feature vectors had been used to identify ham and phish emails.

EXPERIMENTAL RESULTS

The result of the proposed fast detection method was evaluated using TP, FN, TN, FP and accuracy. The neural network model was trained and tested using with different number of data samples data for both features vectors (with 17 and 12 features).

The neural network with 17 features: The system was trained and tested using 8801 vectors (emails sample), the train vectors was 587 (66 ham and 521 phish vectors), the test vectors was 8214 vector (4430 ham and 3784 phish). Different learning rate, momentum and hidden nodes have been investigated to find the best neural network setting values. The best learning rate was 0.4 and the momentum was 0.07. Table 2 shows the detection results for feed forward neural network has one hidden layer consist of 20 nodes and trained using with learning rate = 0.4 and momentum = 0.07.

The neural network with 12 features: The system was trained and tested using total 8697 vectors (emails sample), the train data was 282 (21 ham and 261 phish) vectors, the test data was 8415 (4415 ham and 4000

Table 2: The detection results for the case of using 17 discriminating features

No. of hidden nodes	TP (%)	TN (%)	FN (%)	FP (%)	Acc.	Training time (msec.)	Testing time (msec.)	Average test time for one email (msec.)
1	99.94	90.47	0.05	9.52	95.21	0.73	5.41	0.0006
2	99.86	99.72	0.13	0.27	99.80	0.50	6.98	0.0007
3	99.86	99.81	0.13	0.18	99.84	0.61	8.52	0.0009
4	100	99.81	0	0.18	99.91	0.75	10.55	0.0011
5	100	99.81	0	0.18	99.91	0.82	11.59	0.0013
6	100	99.81	0	0.18	99.91	0.89	12.62	0.0014
7	99.97	99.75	0.02	0.24	99.87	0.99	13.46	0.0015
8	99.97	99.75	0.02	0.24	99.87	1.12	15.61	0.0017
9	99.97	99.75	0.02	0.24	99.87	1.27	17.89	0.0020
10	99.97	99.75	0.02	0.24	99.87	1.32	18.45	0.0020
11	99.97	99.75	0.02	0.24	99.87	1.55	21.32	0.0024
12	99.97	99.75	0.02	0.24	99.87	3.89	22.07	0.0025
13	99.97	99.75	0.02	0.24	99.87	1.70	23.48	0.0026
14	99.97	99.75	0.02	0.24	99.87	1.84	25.04	0.0028
15	99.97	99.70	0.02	0.29	99.84	1.85	26.48	0.0030
16	99.94	90.63	0.05	9.36	95.29	2.01	27.51	0.0031
17	99.97	99.75	0.02	0.24	99.87	2.11	28.86	0.0032
18	99.94	90.63	0.05	9.36	95.29	2.26	31.71	0.0036
19	99.94	90.63	0.05	9.36	95.29	2.26	31.99	0.0036
20	99.94	90.63	0.05	9.36	95.29	2.47	34.70	0.0039

Table 3: The detection results for the case of using 12 discriminating features

No. of nodes	TP (%)	TN (%)	FN (%)	FP (%)	Acc.	Training time (msec.)	Test time (msec.)	Average test time for one email (msec.)
1	99.65	96.03	0.35	3.96	97.84	0.79	5.08	0.0006
2	99.75	99.90	0.25	0.09	99.83	0.27	6.68	0.0007
3	100	99.90	0	0.09	99.95	0.32	7.61	0.0009
4	100	5.20	0	94.79	52.60	0.36	8.69	0.0010
5	100	5.20	0	94.79	52.60	0.41	10.06	0.0011
6	100	5.20	0	94.79	52.60	0.42	11.08	0.0013
7	100	5.20	0	94.79	52.60	0.48	12.50	0.0014
8	100	5.20	0	94.79	52.60	0.53	13.59	0.0016
9	100	5.20	0	94.79	52.60	0.53	14.78	0.0017
10	100	5.20	0	94.79	52.60	0.63	16.46	0.0019
11	100	5.20	0	94.79	52.60	0.63	17.96	0.0021
12	100	5.20	0	94.79	52.60	0.72	18.83	0.0022
13	100	5.20	0	94.79	52.60	0.72	19.59	0.0023
14	100	5.20	0	94.79	52.60	0.74	20.24	0.0024
15	100	5.20	0	94.79	52.60	0.79	22.64	0.0026
16	100	5.20	0	94.79	52.60	0.90	24.44	0.0029
17	100	5.20	0	94.79	52.60	0.87	26.36	0.0031
18	100	5.20	0	94.79	52.60	0.99	27.45	0.0032
19	100	5.20	0	94.79	52.60	0.99	28.47	0.0033
20	100	5.20	0	94.79	52.60	1.03	29.38	0.0034

phish) vectors. Different learning rate, momentum and hidden nodes have been used to find the best results to train the neural network. The best learning rate was (0.1) and the momentum was (0.01). Table 3 shows the result for 20 hidden nodes with learning rate = 0.1and momentum = 0.01.

The time of training and testing the neural network was, also, calculated for both 17 features and best 12 features.

The results listed in Table 2 shows that the accuracy was the same with 4, 5 and 6 hidden neurons, it also shows that as the number of neurons in the hidden layer increase the time taken to train the neural increases so the best result was when the number of hidden nodes is taken 4. The results listed in Table 3 shows that the best accuracy was when 3 hidden neurons is taken. As the number of nodes increase above 3 neurons the accuracy drop down to 50%.

CONCLUSION

The proposed method of training based on neural network had achieved very low FN and low FP for both cases (i.e., using 17 features or only the best 12 features). The extraction of only the best features caused time drop in testing single email to be 0.81% and the number of neurons (nodes) in the hidden layer to be 3 nodes only which is then the required number in case of using 17 features. The best achieved classification result (when using 12 features) was 99.95%.

REFERENCES

Aaron, G., R. Rasmussen and A. Routt, 2014. Global Phishing Survey: Trends and Domain Name Use in 1H2014. Anti Phishing Working Group (APWG), Lexington.

Aaron, G. and R. Rasmussen, 2015. Global Phishing Survey: Trends and Domain Name Use in 2H2014. Anti Phishing Working Group (APWG), http://www.apwg.org ○ info@apwg.org.

Aggarwaly, A., A. Rajadesingan and P. Kumaraguru, 2012. PhishAri: Automatic Realtime Phishing Detection on Twitter. Proceedings of the IEEE eCrime Researchers Summit (eCrime), Las Croabas, pp: 1-12.

Almomani, A., B.B. Gupta, S. Atawneh, A. Meulenberg and E. Almomani, 2013. A survey of phishing email filtering techniques. IEEE Commun. Surv. Tutorials, 15(4): 2070-2090.

Chandrasekaran, M., R. Chinchani and S. Upadhyaya, 2006. PHONEY: Mimicking user response to detect phishing attacks. Proceedings of the IEEE International Symposium on a World of Wireless Mobile and Multimedia Network (WoWMoM), Buffalo-Niagara Falls, NY, USA, pp: 672-676.

Fette, I., N. Sadeh and A. Tomasic, 2007. Learning detection phishing emails. Proceedings of the International World Wide Web Conference Committee (IW3C2), Banff, Alberta, Canada, pp: 649-656.

Ghazhie, N. and L.E. George, 2013. Methodologies to Detect Phishing Emails. Published by Scholars' Press. Retrieved from: http://dl.acm.org/ citation.cfm?id=2613502.

Kavada, A., 2007. The European Social Forum and the Internet: A Case Study of Communication Networks and Collective Action. PHD Thesis, University of Westminster, London, United Kingdom.

Khonji, M., Y. Iraqi and A. Jones, 2011. Lexical URL analysis for discriminating phishing and legitimate e-mail messages. Proceedings of the IEEE 6th International Conference Internet Technology and Secured Transactions (ICITST), Abu Dhabi, United Arab Emirates, pp: 422-427.

Khonji, M., Y. Iraqi and A. Jones, 2012. Enhancing phishing e-mail classifiers: A lexical URL analysis approach. Int. J. Inform. Secur. Res., 2(1/2): 236-245.

Tally, G., R. Thomas and T.V. Vleck, 2004. Anti-Phishing: Best Practices for Institutions and Consumers. White Paper, McAfee Inc.

Vaishnaw, N. and S.R. Tandan, 2015. Development of anti-phishing model for classification of phishing e-mail. IEEE Int. J. Adv. Res. Comp. Commun. Eng., 4(6): 39-45.