DOI: http://dx.doi.org/10.26483/ijarcs.v9i1.5471

ISSN No. 0976-5697

Volume 9, No. 1, January-February 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A SIMILARITY-BASED FRAMEWORK FOR DETECTING PHISHING WEBSITES

Hossain Kordestani
Dept. of computer engineering and information technology
Amirkabir University of Technology
Tehran, Iran

Mehdi Shajari
Dept. of computer engineering and information technology
Amirkabir University of Technology
Tehran, Iran

Abstract: Phishing, which is categorized as social engineering, is turning to a hotbed for modern fraudulence. Since the first phishing attack in the 90s, many solutions were suggested. List-based methods are a common solution, mostly in commercial methods; although they are prone to zero-day attacks. Researchers introduced many methods to detect the phishing by its properties. In this paper a novel framework is introduced, which is based on textual similarity of the phishing website to the original one; this scans the possible meant-to-visit websites and by calculating the similarity, decides the destination. This framework was implemented and tested against real-world websites and provides decent accuracy. This framework not only detects the phishing can guide the user to the original website.

Keywords: Security, Phishing, Internet Security, Anti-Phishing, Textual Properties

I. INTRODUCTION

As the use of electronic services such as e-banking, e-mail and e-commerce grow; more attempt to steal the identity of the users. Phishing is a fraudulent act of deceiving the users by the looks of a legitimate website to steal their credentials. The phishing attacks are mostly ignited by an email or a text message, consisting a threat or a seductive offer to lure the victim to the phishing website.

According to the latest statistics, in January 2018 one in 2,836 emails lead to phishing websites [1]. Most of the attacks, targeting financial institution and payment services which form the 61% of around 290,000 unique phishing attacks in the first quarter of the year 2017[2]. In six months internet users lost about 687 million dollars to the phishing attacks, which have 31% increase in the similar period in the previous year. [3] Therefore it becomes imminent to develop a fast and precise phishing detection tool.

Since the first phishing attack in the 90s, many solutions were suggested to mitigate them. List-based methods are simple common solutions, as most commercial methods take list approach; although they are prone to zero-day attacks. Researchers introduced many methods to detect the phishing by its properties; most of them rely on the popularity and search-engine results for detection. That causes falsely detecting low-profile legitimate websites.

In this paper, a novel framework was introduced which is relied on the similarity of the websites. As they are not associated with the popularity, therefore low-profile websites are not to be detected. Since a phishing webpage tries to lure the victim to mistakes the visiting website with a legitimate one, therefore it has to be similar to the legitimate website. In this paper, in addition to the framework, an implementation of the framework is introduced. The implementation uses textual-properties for the calculating the similarities.

This framework operates using three components: 1) modeling component which is to model the website in order to make comparison possible. 2) Comparison component, which takes the models of websites and returns the similarity ratio of them. 3) Candidate extraction component, which reduces the search area for finding the legitimate website.

The implementation of the framework uses textual properties extraction and comparison for the first two components and for the third components uses exploring links and keywords. In [4], it is shown around 85 percent of phishing websites have a link to the original website, therefore exploring the links appears a suitable solution.

The remainder of the paper is organized as follows: In section II a number of related works are described; section III presents the architecture of the detection system. Section IV discusses the detection framework. Section V presents the manner of test and experiment. Section VI shows the effect of configuration, section VII discusses the results of the experiment and section VIII is the discussion of the results and section IX is the conclusion and future works.

II. LITERATURE REVIEW

The methods to mitigate phishing attacks generally take one of the following perspectives:

1. Mitigating the reasons people can be deceiving and are prone to the phishing attacks by social and human studies
2. Introducing a tool for detecting the phishing
3. Training the users to be resistant to the attacks

This paper follows the 2nd perspective. These group of methods can be categorized as follow:

A. *List-Based Methods*

These types of methods use a list of websites of clear state of phishing or clean, and the visited website is checked against the list. These methods can have two kinds of lists:

a. White List: These methods utilize the list of legitimate web pages, which can be formed from the users browsing history and considers the new website a possible threat; and alert as phishing with a computational probability. [4] and [5] are two of the methods with this approach.

b. Black List: These methods, on the contrary, use lists of phishing websites. [6], [7]and [8] use this approach. Many commercial tools including browsers and security toolbars have this approach.

These methods have low computational overhead, and when the visited website is in the lists, have a perfect accuracy on the decision; but when the website is new to the list, the

correctness drops. Therefore, these methods are helpless against zero-day phishing attacks.

B. User-Based Methods

User-based methods combine the three perspectives and provide useful information to the user, and let him decide the action. Though, these methods might be able to detect zero-day attacks; but un-informed users would never take the cautions seriously. [9] is an example of this method.

It's not recommended to have users decide because he usually doesn't have enough background knowledge to make the correct decision.

C. Website Analysis Method

These methods use analysis of the websites and regarding the properties of the phishing websites and legitimate ones, to make the detection. The analysis can be in the following parts of the website:

a. Content: These methods study the contents of the website. [10], [11], [12], [13], [14], [15], [16], [17] and [18] introduce methods which will be grouped as content analysis methods.

b. Communication: These methods analyze the surroundings. [19] uses the communications to identify the existence of databases near the website, which might be used for collecting stolen data. [20] and [21] scans the connections of the website to analyze the communication with original website, which is a common parameter in most phishing websites.

c. Search Engine Based: These methods rely on the information provided by search engines and other online sources for the detection. [22]and [23] are of this kind. These techniques are prone to search engine optimization acts.

The key challenge in this type of tools is to select the most precise and optimum attributes. The main goal is to have high detection rate, i.e. low false positive and false negative.

III. SYSTEM ARCHITECTURE

Figure 1 shows the overview of the system architecture. This framework is based on textual properties of the website; the key feature of websites in use is the fact phishing websites have textual similarity to the original website. The framework consists of three components:

- Textual Properties Extractor: This component is to analyze the website and extract textual properties of it.
- Candidates Finder: This component provides a list of websites which are a possible target of the victim.
- Compare Unit: this component, using both textual properties of visiting and candidate websites; compares them

These components are thoroughly discussed in the next section.

IV. A SIMILARITY BASED ANTI-PHISHING FRAMEWORK

Phishing which is a malicious act of stealing valuable information, by masquerading a website; have to lure the user into the trap. Therefore, phisher send an email, or text, to the victim; and tempting or threatening him to enter the website, and victim follows the bait and goes to the website. If the website is similar to the legitimate one, then he might fall into the trap; but in case of difference, he might get suspicious and fly out.

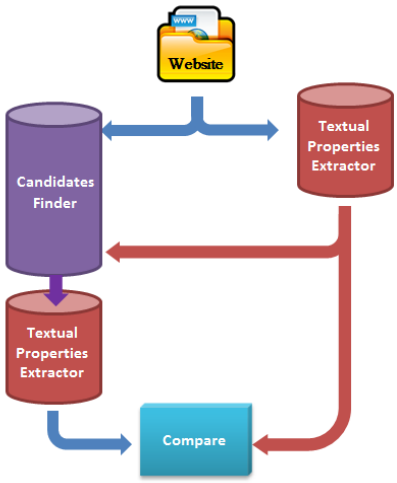

Figure 1 System Architecture

Therefore, the phisher tries to make the phishing website as similar as possible to the legitimate one to make sure that the victim won't get suspicious.

This framework triggers this point of the attack and tries to detect phishing websites by this. That is this framework checks through a list of candidates and analyzes the similarity by both textual properties. On the other hand, the legitimate website is mostly unique.

The following explains the components:

A. Modeling

This component tries to model the website; that is it extracts several properties of the website and make a sound and complete association between a website and its properties. In this case, the set of properties can be called a model. The model of the website in contrast to the website is easy to deal with, therefore the website is modeled before getting into the comparison component.

B. Candidates Extraction

The framework tries to find a similar website to the visiting one and try to detect phishing. As going through all of the internet is impossible therefore it searching region should be narrowed.

This component provides possible websites the victim meant to visit; i.e. a limited number of website which might be similar to the visited ones are extracted here.

This component may use the model of the website or the website itself. The outputs of this component are analyzed to find the intended website.

C. Comparison

This component compares two websites. Since a website is a complex entity, modeling is needed. The comparison algorithm is highly dependent on the modeling algorithm.

After the result of the comparison, it's up to the decision maker, whether to direct the victim of phishing to the safe original website or only alert the user.

V. EXPERIMENTAL METHOD

A. An Implementation of the Framework

For testing the framework, it was implemented. The selected platform was Java. The three components of the framework were implemented as follow:

- Modeling: The modeling component was implemented by extracting the textual property of the website.

AlchemyAPI was used for keyword extraction. The keywords of the body of the websites and title were extracted separately and their intersection was used to extract the brand name.

- Candidates Extraction: Studies on current phishing websites show around 85 percent of them have a link to the original website [4]. Therefore exploring the links and actions of the visited website is a suitable choice. For further candidates, some of the bold textual properties were searched via Google and their results were added to the list. The websites within the domain of visiting are excluded.
- Comparison: For comparing the textual-similarity of the candidate to the visited website, their textual properties were extracted and the trio was respectively compared using string-comparison algorithms and the results of each comparison were put together according to their significance.

B. *Evaluation Metrics*

In detection systems, it's important to hold both false positive and false negative as low as possible. The F1 score is a suitable measure which put both into consideration.

Similar to other detection systems, the main criteria is false positive and false negative. Using F1 score which holds both criterions ease the evaluation process

C. *Dataset Sources*

PhishTank updates its database regularly with the help of its users, and hold a collection of more than 10,000 verified online phishing websites[24]. Therefore it was used for phishing websites collection.

Alexa provides a list of most one million websites; which the bottom of the list is suitable choices for low profile websites; additionally, Google, Ad-Planner, and Yahoo Directory also provide high-profile websites. A combination of both was used for the test.

In order to evaluate the system, a dataset of websites consisting random selection of the data sources was selected; these had around 100 websites from each source.

D. *Evaluation Method*

For evaluating phishing detection system, there are two common methods. Random-Based evaluation and Time-Based Evaluation. The former evaluation the overall performance of the system; and the latter evaluates the performance under real-world circumstances. [25]

In this paper, the proposed system is evaluated using the random-based method; in which a random selection of legitimate and phishing websites for both training and testing phase is done.

VI. EXPERIMENTING THRESHOLDS

This framework is based on textual similarity and if the similarity becomes more than a threshold it detects this as a phishing; in this part, the effect of this parameter is shown. For showing how threshold can affect the detection rate, the implemented framework was tested for different thresholds. Figure 2 illustrates how threshold can change the detection rate.

This experiment shows, as the threshold gets higher, that is more similarity is required to be marked as analogous, the true negative gets higher, and on the other hand, more phishing website get out of detection system and therefore less true positive.

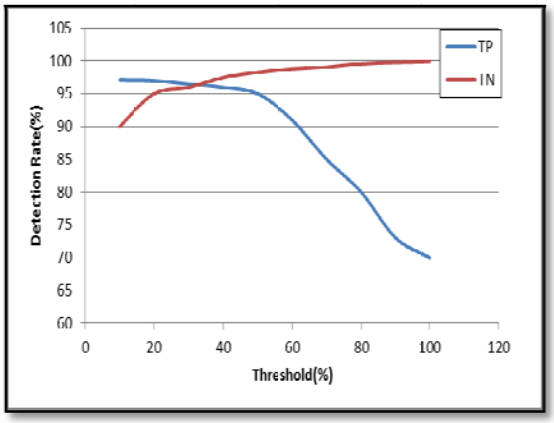

Figure 2 Effect of Threshold in FP and FN

This method shows the threshold value should be valued with making a compromise between false positive and false negative; because it's impossible to lower both.

VII. COMPARISON AND DISCUSSION

In this part, this implemented version of framework compared to two other detection systems. For this purpose, a dataset consisting random selection of the sources were used for this purpose.

Cantina, based on TF-IDF keyword extraction uses a few heuristic parameters for detection [17].

Cantina+ uses a dozen of parameters for detection. It also uses a collection of a hashed version of phishing bodies; to detect phishing website which is done with specific tools [25]. Because the latter is categorized as list-based methods it is not included in the implementation. The result of this comparison is shown in Table I.

Table I Comparison with similar methods

Method	True Positive	False Positive	F-Measure
Cantina	51.9%	25.8%	58.4%
Cantina+	97.6%	23.7%	88.3%
Implemented Framework	96.2%	2.1%	97.0%

The outcome of the evaluation suggests the detection rate of the implemented framework is similar to Cantina+, but since the framework doesn't count on the popularity of the website, it doesn't falsely detect low-profile websites; and therefore the false positive rate is so much lower in the proposed method.

The true positive of the both implemented framework and Cantina+ are high and approximately the same. Cantina+ using detailed features can detect phishing accurately and also the framework using the similarity, which all of the phishing websites include can also detect phishing websites precisely.

False positive in the implemented framework is low because it doesn't rely on the credibility of the website nor on its page rank and similar parameters related to the website popularity; therefore the low profile legitimate websites are not falsely detected as phishing; that makes the false positive very low. On the other hands the other two, which detects partly based on the popularity of the websites; are prone to false detection.

The main source of false positive in Cantina and Cantina+ are the low-profile websites; these are less popular therefore the detection system misinterpret their low popularity as phishing.

It worth mentioning the majority of the false detection are because of the implementation, that is for falsely detecting as

phishing, the suggesting domain is another address of the same website; therefore the main source of false positive is the problem of distinguishing different domains of a website, which are similar.

False negative is mainly because of the websites with too little of texts, which makes the textual property of the website rather empty. That is, the modeling is done incorrectly. E.g. website is dedicated for login; these pages include only an HTML form and the text is limited to general keywords such as username, login and etc.; which can't be used as decision guide.

Another source of false decision, both false positive and false negative, is the language of the website. As in the implementation of the framework, the property extractor doesn't implement some of the languages; therefore the textual properties and the decisions are not reliable. Albeit, this is a drawback of the implementation rather than in the framework.

VIII. DISCUSSION

As mentioned in previous sections, the phisher tries to make the phishing website as similar as possible to the legitimate one to make sure that the victim won't get suspicious. That is this framework checks through a list of candidates and analyzes the similarity. On the other hand, the legitimate website is mostly unique.

In website-analysis phishing detection methods, the properties of websites are used for detection; these properties can rely on the URL, the connections, the search engine and other similar information. This information is proven to be similar to some of the legitimate websites. For example, page rank and results in a search engine can detect phishing website; but for the low-profile website, the rank is also low, which causes a large false positive.

Legitimate websites are unique; because each has a different taste and service behind it; that stops the framework from falsely detecting legitimate websites; therefore the false positive is very low in this implementation.

As discussed above, the similarity is a key feature in the phishing websites, and as this framework uses this feature for detection, the false negative would also be very low.

IX. CONCLUSION

In this paper, a novel framework was introduced which is relied on textual-properties. These are not associated with the popularity, and therefore low-profile websites are not to be detected.

This framework was implemented by using the textual property for modeling. The implementation was used for evaluation of the framework; although it has some drawbacks, and can't identify websites which has little words on it. That is when phisher doesn't put much information on the website; or he can put the information in the form of images, which doesn't provide a text or the text consist of general terms. In this case, this implementation of the framework can't extract textual properties which this system is based on; therefore it can't be processed and the detection can't perform. But it's not a drawback of the framework.

For websites, which has limited texts, another implementation can be used to improve the results. For future works, the graphical properties can be added to the detection; to make it harder to entice by the attackers

X. REFERENCES

[1]. Symantec Intelligence. (2018, January) Symantec. [Online]. https://www.symantec.com/security_response/publications/monthlythreatreport.jsp

[2]. APWG. (2017, October) Anti-Phishing Working Group. [Online]. http://docs.apwg.org/reports/apwg_trends_report_h1_2017.pdf

[3]. RSA. (2012, August) RSA. [Online]. http://blogs.rsa.com/rsafarl/phishing-in-season-a-look-at-online-fraud-in-2012/

[4]. Y. Cao, W. Han, and Y. Le, "Anti-Phishing based on automated individual white-list," Proceedings of the 4th ACM workshop on Digital identity management, pp. 51-60, 2008.

[5]. N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell, "Client-side defense against web-based identity theft," in Client-side defense against web-based identity theft, San Diego, 2004.

[6]. R. Dhamija and J. D. Tygar, "The battle against phishing: Dynamic Security Skins," in Proceedings of the 2005 symposium on Usable privacy and security, New York, 2005, pp. 77-88.

[7]. J. Kang and D. Lee, "Advanced White List Approach for Preventing Access to Phishing Sites," in Convergence Information Technology, 2007, pp. 491-496.

[8]. M. Sharifi and S.H. Siadati, "A phishing sites blacklist generator," in IEEE/ACS International Conference, 2008, pp. 840-843.

[9]. D. Miyamoto, H. Hazeyama, and Y. Kadobayashi, "SPS: A Simple Filtering Algorithm to Thwart Phishing Attacks," in Technologies for Advanced Heterogeneous Networks, Berlin, 2005, pp. 195-209.

[10]. B. Adida, S. Hohenberger, and R. L. Rivest, "Fighting Phishing Attacks : A Lightweight Trust Architecture for Detecting Spoofed Emails," in In DIMACS Wkshp on Theft in E-Commerce., 2005.

[11]. M. Dunlop, S. Groat, and D. Shelly, "GoldPhish: Using Images for Content-Based Phishing Analysis," in 2010 Fifth International Conference Internet Monitoring and Protection, 2010, pp. 123-128.

[12]. S. Nakayama, H. Yoshiura, and I. Echizen, "Preventing False Positives in Content-Based Phishing Detection," in Intelligent Information Hiding and Multimedia Signal Processing, 2009., 2009, pp. 48-51.

[13]. S. S. Tseng, K. Y. Chen, T. J. Lee, and J. F. Weng, "Automatic content generation for anti-phishing education game," in Electrical and Control Engineering, 2011, pp. 6390-6394.

[14]. B. Wardman, T. Stallings, G. Warner, and A. Skjellum, "High-performance content-based phishing attack detection," in eCrime Researchers Summit, 2011, pp. 1-9.

[15]. C. Whittaker and B. Ryner, "Large-Scale Automatic Classification of Phishing Pages," in 17-th Annual Network and Distributed System Security Symposium, 2010.

[16]. Jianyi Zhang et al., "An content-analysis based large scale Anti-Phishing Gateway," in Communication Technology (ICCT), 2010 12th IEEE International Conference on, November 2010, pp. 979-982.

[17]. Y. Zhang, J. I. Hong, and L. F. Cranfor, "Cantina: a content-based approach to detecting phishing web sites," in Proceedings of the 16th international conference on World Wide Web, New York, 2007, pp. 639-648.

[18]. H. Zhang, G. Liu, T.W. S. Chow, and W. Liu, "Textual and Visual Content-Based Anti-Phishing: A Bayesian Approach," IEEE Transactions on Neural Networks, vol. 22, no. 10, pp. 1532-1546, October 2011.

[19]. G. Liu, B. Qiu, and L. Wenyin, "Automatic Detection of Phishing Target from Phishing Webpage," in Pattern Recognition (ICPR), 2010 20th International Conference on,

2010, pp. 4153-4156.

[20]. C. Yue and H. Wang, "BogusBiter: A transparent protection against phishing attacks," ACM Transactions on Internet Technology , vol. 10, no. 2, pp. 6:1-6:31, May 2010.

[21]. Y. Pan and X. Ding, "Anomaly Based Web Phishing Page Detection," in Computer Security Applications Conference, 2006, pp. 381-382.

[22]. J.H. Huh and H. Kim, "Phishing Detection with Popular Search Engines : Simple and Effective," FPS'11 Proceedings of the 4th Canada-France MITACS conference on Foundations and Practice of Security, pp. 194-207, 2012.

[23]. H. Kordestani and M. Shajari, "An entice resistant automatic phishing detection," The 5th Conference on Information and Knowledge Technology, Shiraz, 2013, pp. 134-139.

[24]. OpenDNS. PhishTank. [Online]. http://www.phishtank.com

[25]. Guang Xiang, Jason Hong, P. Carolyn Rose, and Lorrie Cranor, "CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites," ACM Transactions on Information and System Security, vol. 14, no. 2, September 2011.

[26]. Google. (2011, July) Top 1000 sites - DoubleClick Ad Planner. [Online]. http://www.google.com/adplanner/static/top1000/

[27]. Christopher J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining and Knowledge Discovery, vol. 2, no. 2, pp. 121-167, June 1998.