Detection of Phishing Websites using Machine Learning

Conference Paper · October 2020

DOI: 10.1109/IEEECloudSummit48914.2020.00022

6 authors, including:

Abdul Razaque
Gachon University
203 PUBLICATIONS **1,843** CITATIONS

SEE PROFILE

Mohamed Ben Haj Frej
University of Bridgeport
26 PUBLICATIONS **166** CITATIONS

SEE PROFILE

Fathi Amsaad
Eastern Michigan University
87 PUBLICATIONS **527** CITATIONS

SEE PROFILE

Detection of Phishing Websites using Machine Learning

[1]Abdul Razaque
Computer Engineering and Telecommunication Department
International IT University, Almaty
Kazakhstan
[1]a.razaque@iitu.kz

[2]Mohamed Ben Haj Frej
Department of Computer Science And Engineering
University of Bridgeport
Bridgeport, CT, USA
[2]mbenhaj@bridgeport.edu

[3]Dauren Sabyrov
Computer Engineering and Telecommunication
International IT University
Almaty, Kazakhstan
[3]daukeke99@gmail.com

[4]Aidana Shaikhyn
Computer Engineering and Telecommunication
International IT University
Almaty, Kazakhstan
[4]aidana759@gmail.com

[5]Fathi Amsaad
School of Information Security and Applied Computing (SISAC)
Eastern Michigan University
Ypsilanti, MI, USA
[5]famsaad@emich.edu

[6]Ahmed Oun
Department of Electric Engineering and Computer Science
University of Toledo
Toledo, OH, US
[6]Ahmed.oun@rockets.utiledo.edu

Abstract
Phishing sends malicious links or attachments through emails that can perform various functions, including capturing the victim's login credentials or account information. These emails harm the victims, cause money loss, and identity theft. In this paper, we contribute to solving the phishing problem by developing an extension for the Google Chrome web browser. In the development of this feature, we used JavaScript PL. To be able to identify and prevent the fishing attack, a combination of Blacklisting and semantic analysis methods was used. Furthermore, a database for phishing sites is generated, and the text, links, images, and other data on-site are analyzed for pattern recognition. Finally, our proposed solution was tested and compared to existing approaches. The results validate that our proposed method is capable of handling the phishing issue substantially.

Keywords: Phishing attack, Semantic analysis methods, Database, Machine Learning

I INTRODUCTION

With the widespread usage of the Internet for online banking and trade, phishing attacks and forms of identity theft-based scams are becoming extremely popular among the hacker communities. In 2004 alone, more than 50 million phishing emails were sent. Their result was 10 billion dollars of damage to banks and financial institutions [1]. Most of the recent phishing attacks are carried out as a three-step process. In the first step, the phishers send emails to their victims from social engineering attacks, webpages, and forums. Large volumes of phishing emails with legal banking domains are sent out using anonymous servers or compromised machines. These emails contain hyperlinks with an appearance similar to the legitimate website. The fake webpage contains input forms requesting personal critical information such as credit card, social security numbers, mother's maiden name, etc. Although existing spam filtering techniques can be employed to combat phishing emails, these measures are not entirely scalable. Several readily available tools can bypass both the statistical and rule-based spam filters [2]. As these mechanisms are not uniquely tuned for the detection of phishing emails despite their existence, the threats caused by phishing emails are prevalent [3]. Furthermore, unlike spamming, which impacts bandwidth, phishing attacks directly affect their victims by inflicting a hefty loss due to monetary damage [4].

Moreover, attackers can use technical vulnerabilities to construct socially engineered messages (i.e., use of legitimate, but spoofed, domain names can be far more persuasive than using different domain names), which makes phishing attacks a severe problem. Effective mitigation would require addressing issues at the technical and human layers. Since phishing attacks aim at exploiting weaknesses found in humans (i.e., system end-users), it is difficult to mitigate them. For example, as evaluated, end-users failed to detect 29% of phishing attacks, even when trained with the best performing user awareness program [5]. On the other hand, software phishing detection techniques are evaluated against phishing attacks, which makes their performance practically unknown with targeted forms of phishing attacks. These limitations in phishing mitigation techniques have almost resulted in security breaches against several organizations, including leading information security providers [6].

More specifically, we highlight the main contributions as follows:

- The novel extension of the Google Chrome web-browser is based on Blacklisting and semantic analysis methods that will be integrated successfully to efficiently identify and prevent the phishing attack.
- IP URLs and redirection of the user's information are checked successfully using phishing detection. User's redirection algorithms are proposed.

The rest of the paper is organized as Section II; discusses the salient features of existing approaches in the related work, Section III; problem identification and significance, and Section IV which shows the implemented

algorithms. Section V is the implementation. Finally, the entire paper is concluded in section VI.

II RELATED WORK

In this section, salient features of existing approaches are summarized. Abu-Nimeh et al. [5] compared the predictive accuracy of several machine learning methods, including LR, CART, RF, NB, SVM, and BART. They used 43 features to capture phishing emails. They analyzed 1,718 legitimate emails and 1,117 phishing emails. For the case of Random Forests, their results showed a 7.72% as the lowest error rate. Basnet et al. [6] evaluated six diverse detection methods that are based on machine learning. Using 12 features, they analyzed 3,027 legitimate emails and 973 phishing emails. Their results showed the lowest error rate at 2.01%. For [7] and [8], even though they used different experimental parameters, their detection of phishing emails, based on machine learning, led to high accuracy.

In [9], the solution to the phishing emails problem was provided. It checks the email for 17 elements of phishing emails. Some of them are not suitable in modern life because they don't follow the current trends. For instance, if the URL has '@' sign, it will be interpreted as a phishing attack, or if the text inside the anchor HTML tag is not the same as that in its attribute, it will also be considered as phishing. Still, now to register on some sites, you need to follow the link with the text "Click here to finish the registration." [10] provided the solution of the detection of a phishing attack using neural networks, features of MIME, and checking the context of the email on 6 elements, one of them being the total number of links in the message. Some of the new ad sites send emails with 10 and more links on their products, and they will be considered as phishers even though they are not.

In [11], phishing emails and phishing sites can be detected according to JavaScript functions. If there are functions eval() or exec(), they can be considered as malicious. Still, if the functionality of these functions is overwritten in another way, this detection will not be enough. There is a challenge of detecting phishing sites, as the number of features for detecting phishing sites is less than that of detecting phishing emails. It indicates that the detection of phishing sites is more complicated than that of phishing emails. Thus, our paper mainly focuses on identifying phishing websites and providing a higher accuracy rate.

III PROBLEM IDENTIFICATION

Phishing affects individuals and companies worldwide. It is difficult to track the perpetrators since it is carried out across the borders. In addition, the phishers' method, "fast-flux," uses a large pool of proxy servers and URLs to hide the actual location of the phishing site. Simultaneously, it is more challenging to blacklist the site as the server used requires a lot of work. Phishing attacks are aimed at vulnerabilities that exist in systems due to human factors. Many cyber-attacks spread through mechanisms that exploit weaknesses found in end users, making users the weakest link in the security chain. To solve the problem, different organizations use different methods.

In the Google web store, most anti-phishing extensions are aimed at preventing or avoiding phishing attacks on sites such as Twitter, eBay, Facebook, etc. Another method is based on checking the URL of the site. It divides the text of the URL into distinct segments, and then tries to follow them as links: if the connection is established, the site is considered phishing. This method has many drawbacks, the biggest of which is that it is not adaptive, and the accuracy of their work is very low.

IV PROPOSED PLAN

This section describes the proposed model of phishing attack detection. The proposed model focuses on identifying the phishing attack based on checking phishing websites features and the blacklist database. According to our proposal tool, a few selected features can be used to differentiate between phishing and non-phishing web pages. These selected features include URLs, domain identity, page style and contents, web address bar, and the human social factor. Our paper focuses only on URLs and domain name features. Features of URLs and domain names are checked using several criteria such as IP address, long URL address, redirecting using the symbol "//," and URLs having the mail/mail-to attributes. These features are inspected using a set of rules to select URLs of phishing webpages from the URLs of dangerous websites. The detecting process includes:

- Using a blacklist database, which contains URLs of all phishing websites.
- Using the IP Address: If an IP address is in the URL, such as "http://125.98.3.123/fake.html", users can be sure that someone is trying to steal their sensitive personal information.
- Using mail/mail-to attributes: if these attributes are found in the URL, users can be sure that someone wants to steal their information

Algorithm 1: Phishing detection process

1. **Initialization:**{URL: Uniform Resource Locator ; P_c: Page contents; W_{Pl} : Web page link; W_{ph}: Phishing Websites; M_w : Warning message; DB_{bl} : Blacklist database}
2. **Input:** $\{W_{Pl}\}$
3. **Output:** $\{M_w\}$
4. Set URL
5. If URL = DB_{bl} then
6. Display M_w
7. End-if
8. Else If URL $\neq$ DB_{bl} then
9. Get P_c
10. Obtain $W_{Pl} \in P_c$
11. If URL $\in W_{Pl} = DB_{bl}$ then
12. Display M_w
13. End-if

The graphical representation of the phishing detection process is depicted in Figure 1.

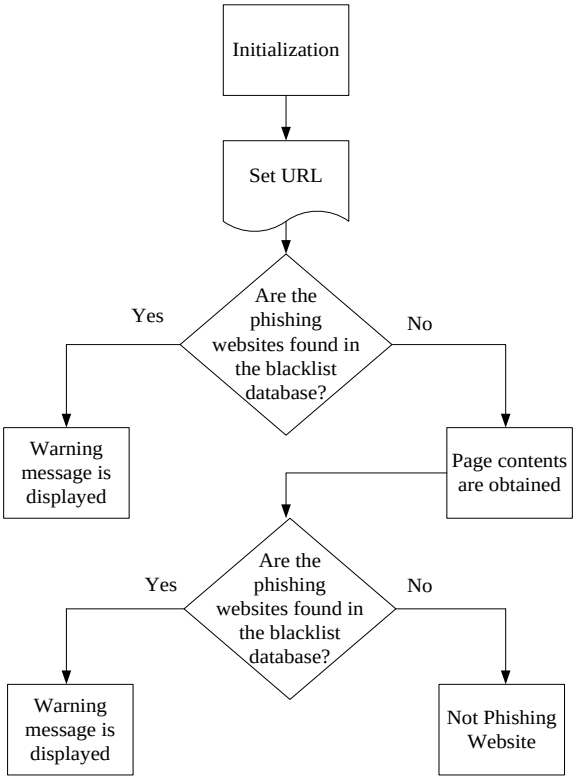

Figure 1: Visualization of the phishing detection process

Algorithm 1 describes the phishing detection process. In step 1, used variables are initialized. Steps 2-3 show the input and output processes, respectively. In step 4, the URL is set to detect the blacklist websites. In steps 5-6, the blacklist website detection process is conducted if blacklist websites are found, then a warning message is displayed. Steps 8-9 show if the URL is not found in the blacklist database, then the page contents are obtained. Steps 10-12 illustrate the process of getting the web page links from the page contents. If web the page links are part of the blacklist database, then the message is displayed.

Algorithm 2: Checking URL for IP address

1. **Initialization:** {URL: Uniform Resource Locator ; IP: IP address; W_P : Web page; M_w : Warning message; }
2. **Input:** {URL }
3. **Output:** $\{M_w\}$
4. Get URL
5. Check URL
6. If IP ∈ URL, then
7. Display M_w

End-if

Algorithm 2 describes the checking of the URL for the IP address process. In step 1, used variables are initialized. Steps 2-3 show the input and output processes, respectively. In steps 4-5, URL is got and checked for IP address. Steps 6-7 show if the IP is found in the URL, then the message is displayed. The graphical representation of checking the URL for the IP address process is depicted in Figure 2.

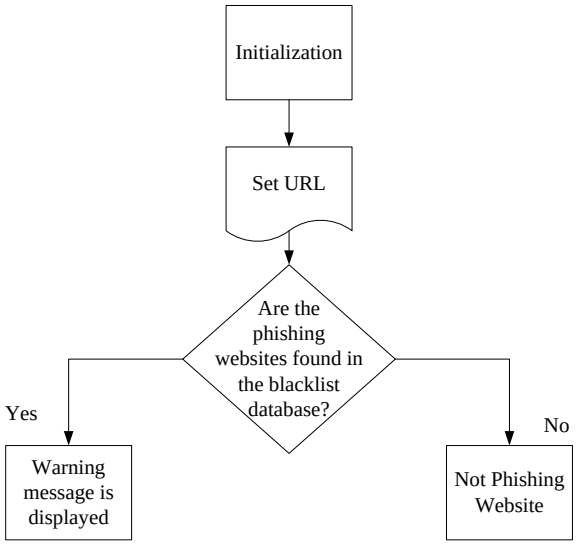

Figure 2: Visualization of checking URL for IP address

Algorithm 3: Checking redirection of the user's information

1. **Initialization:** {URL: Uniform Resource Locator; P_c: Page contents; S_b: Submit buttons; M_w : Warning message; T_m : mailto/mail attributes}
2. **Input:** $\{P_c\}$
3. **Output:** $\{M_w\}$
4. Obtain $S_b \in P_c$
5. If $T_m \in S_b$ then
6. Display M_w
7. End-if

Algorithm 3 describes checking redirection of the user's information process. In step 1, used variables are initialized. Steps 2-3 show the input and output processes, respectively. In step 4, the process of obtaining submit buttons from the page content is demonstrated.

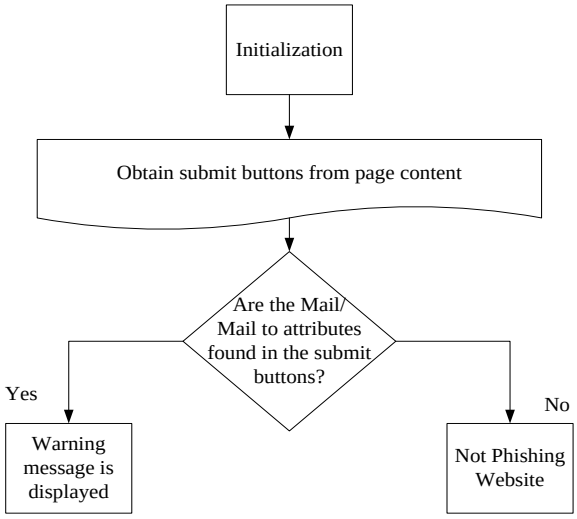

Figure 3: Visualization of checking redirection of the user's information

In steps 5-6, if mail-to/mail attributes are found in submit buttons, then a warning message is displayed. The graphical representation of checking redirection of the

user's information process is shown in Figure 2.

V IMPLEMENTATION AND RESULTS

To solve the phishing problems, we developed the extension for the Google Chrome web browser. In the development of the extension, we used JavaScript PL. To identify and prevent the phishing attack, we used a combination of Blacklisting and semantic analysis methods. The blacklisting method means that we use the database of phishing sites, and if the user follows one of such sites, the corresponding message will be shown. In the second method, we create our URLs database of phishing sites and update them after each meeting with such sites. After that, if the URL of the last site is not found in the database, our extension checks email, its text, links, and images for patterns of phishing sites. Parameters used for implementation are given in Table 1.

TABLE 1: Showing and describing the used equipment

Equipment	Specification
Programming language	JavaScript
Graph showing	MS Word
Data showing	Excel
Operating system	Linux
Memory	1Tb HDD
RAM	8 GB DDR3
Used software	Google Chrome Web Browser, Microsoft visual studio code

- Phishing website detection
- CPU consumption
- Accuracy

A. Phishing website detection

After developing our proposed APE, we compared the 'Anti-phishing and Authenticity Checker' and 'Stop Phishing' extensions for in-depth evaluation. The data visualization and experiment results of the blacklist database depicted in Figure 4.

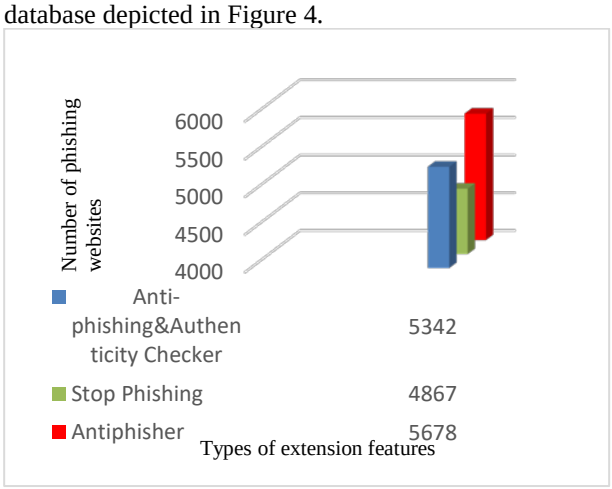

Figure 4. Showing the number of phishing websites in the blacklist

Figure 4 depicts the number of phishing websites in the blacklist database of each extension, as shown. The database consists of URLs that are dangerous for your system.

B. CPU Consumption

The data visualization and experiment results of CPU consumption are depicted in Figure 5. In this experiment, our proposed Anti-Phisher Extension (APE) is compared with similar types of phishing detection approaches, such as phishing for Phishers (FP) [14], Phishing Tweet Detection (PTD) [15], and Remove-Replace Feature Selection (RRFS) [16]. As illustrated in Figure 5, our proposed approach (APE) uses the minimal % CPU consumption as compared to other contending approaches. It is proven that the system can work with maximum speed when using our proposed approach for phishing detection as compared to other competing methods.

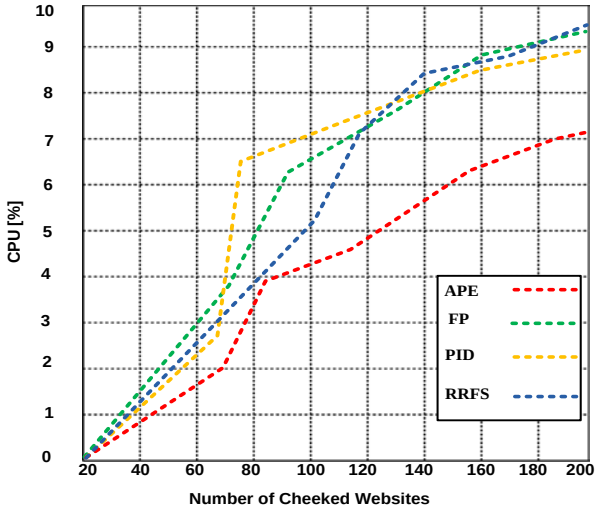

Figure 5: **CPU** Consumption using proposed APE and other contending approaches

C. Accuracy

Here, our proposed APE's accuracy is compared with contending approaches: FP, PID, and PRFS. Figure 6 demonstrates that our proposed approach APE produces a higher efficiency as compared to other methods.

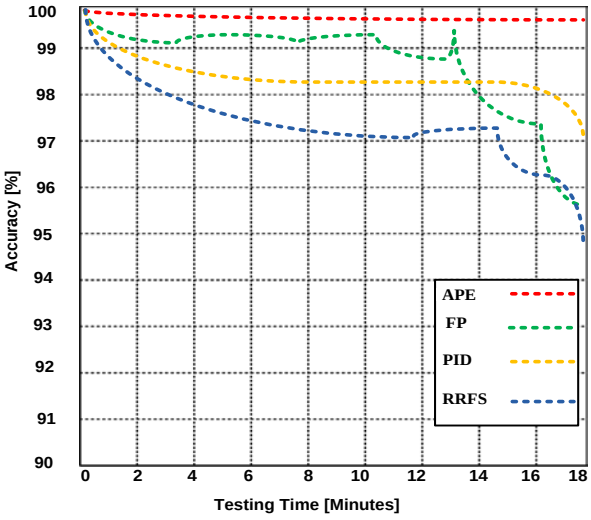

Figure 6: Accuracy of proposed APE and other contending approaches

Our proposed approach's average accuracy has been observed by 99.67%, while other contending approaches fall within 88.5-95.7%. The results prove that our system shows 4-11.4% higher efficiency as compared to other competing methods.

VI CONCLUSION

In this paper, Anti-Phishing Extension has been proposed to handle the phishing contents. The proposed approach consists of three algorithms: 'Phishing detection, checking URL for IP address, and checking redirection of the user's information. This paper focuses on phishing's main consequences, such as stealing personal information from bank accounts, credit cards, social media, etc. as determined by protecting users from phishing attacks to deal with the human factor. The proposed APE approach helps detect phishing attacks efficiently and accurately.

The proposed APE approach works properly with the Google Chrome extension. To show the effectiveness of our implementation, we have programmed in JavaScript language. The results demonstrate the higher accuracy and minimum CPU consumption when using our proposed APE approach, whereas other contending approaches have less accuracy and more CUP consumption. In the future, we plan to obtain additional features by analyzing images and videos and classifying the content of the pages.

REFERENCES

[1] CNET News, *Phishing attacks skyrocket in 2014*, 2014.

[2] Gregory L. Wittel and S. Felix Wu, *On Attacking Statistical Spam Filters*, *First Conference on Email and Anti-Spam*, 2010.

[3] Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2017, October). A comparison of machine learning techniques for phishing detection. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit* (pp. 60-69). ACM.

[4] Bergholz, A., Chang, J. H., Paass, G., Reichartz, F., & Strobel, S. (2011, August). Improved Phishing Detection using Model-Based Features. In *CEAS*.

[5] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, "Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions," in Proceedings of the 28th international conference on Human factors in computing systems, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 373–382.

[6] B. Krebs, "HBGary Federal hacked by Anonymous," http://krebsonsecurity.com/2011/02/hbgary-federal-hacked-by-anonymous/, 2011, accessed December 2011.

[7] Xiang, G., Hong, J., Rose, C. P., & Cranor, L. (2011). Cantina+: A feature-rich machine learning framework for detecting phishing web sites. *ACM Transactions on Information and System Security (TISSEC)*, *14*(2), 21.

[8] Fette, I., Sadeh, N.M., Tomasic, A. "Learning to detect phishing emails." In Proceedings of the 16th International Conference on World Wide Web (WWW'07), May 2017.

[9] Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S. "A comparison of machine learning techniques for phishing detection." In Proceedings of eCrime Researchers Summit (eCryme '07), Oct 2010.

[10] Basnet, R., Mukkamala, S., Sung, A.H. "Detection of phishing attacks: A machine learning approach." Studies in Fuzziness and Soft Computing, 226:373–383, 2014.

[11] Lakshmi, V. Santhana, and M. S. Vijaya. "Efficient prediction of phishing websites using supervised learning algorithms." *Procedia Engineering* 30 (2012): 798-805.

[12] Zhang, Ningxia, and Yongqing Yuan. "Phishing detection using neural network." *Department of Computer Science, Department of Statistics, Stanford University, CA, available at: http://cs229. stanford. edu/proj2012/ZhangYuan-PhishingDetectionUsingNeuralNetwork. pdf (accessed April 23, 2016).[Google Scholar]* (2013).

[13] Sahoo, Doyen, Chenghao Liu, and Steven CH Hoi. "Malicious URL detection using machine learning: a survey." *arXiv preprint arXiv:1701.07179* (2017).

[14] Moreno-Fernández, María M., Fernando Blanco, Pablo Garaizar, and Helena Matute. "Fishing for phishers. Improving Internet users' sensitivity to visual deception cues to prevent electronic fraud." *Computers in Human Behavior* 69 (2017): 421-436.

[15] Liew, Seow Wooi, Nor Fazlida Mohd Sani, Mohd Taufik Abdullah, Razali Yaakob, and Mohd Yunus Sharum. "An effective security alert mechanism for real-time phishing tweet detection on Twitter." *Computers & Security* 83 (2019): 201-207.

[16] Hota, H. S., A. K. Shrivas, and Rahul Hota. "An ensemble model for detecting phishing attack with proposed remove-replace feature selection technique." *Procedia computer science* 132 (2018): 900-907.