International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 05 | May - 2022 Impact Factor: 7.185 ISSN: 2582-3930

A Study on: Implementation of Machine Learning in Malicious Emails

Keerthana V,
PG Scholar,
Department of MCA,
Dayananda Sagar College of Engineering,
Bengaluru, Affiliated to VTU

Suma S,
Assistant Professor,
Department of MCA,
Dayananda Sagar College of Engineering,
Bengaluru, Affiliated to VTU

***Abstract*—the main aim of the current study has been to provide a novel tool for detecting phishing attacks and finding a solution to counteract such types of threats. In this article we describe the process of how to develop a Scrum-based implementation of algorithms for automatic learning, Feature Selection and Neural Networks. This tool has the ability to detect and mitigate a phishing attack registered inside the e-mail server. For the validation of the obtained results, we have used the source of information of blacklist of PhishTank, which is a collaborative clearing house for data and information about phishing on the Internet. The conducted proof of concept demonstrated that the implemented feature selection algorithm discards the irrelevant characteristics of electronic mail and, that the neural network algorithm adopts these characteristics, establishing an optimal level of learning without redundancies. It also reveals the functionality of the proposed solution.**

Keywords—Social Engineering, Phishing, Security, Feature Selection, Neural Networks.

I. INTRODUCTION

According to [1] Social Engineering bases its principle on the fact that, in any system, users appear to be the biggest weakness around their security. Additionally, it is based on the natural tendency of people to react predictably in certain situations. With this method, an attacker easily takes advantage of this natural tendency - for example, when we provide financial details to an apparent bank officer - instead of having to find security gaps in computer systems. Therefore, one of the Social Engineering techniques with the strongest impact is called an identity theft attack in commercial transactions, also known as Phishing [2]. This technique has been used by cyber-criminals and it is characterized by trying to acquire confidential information fraudulently.

Within such scenario, several studies have been analyzed [3]–[7] in which all the given proposals have been exposed and where different automatic learning algorithms have been compared. Nonetheless, these type of attacks continue to appear with a higher frequently and with an increasing complexity.

In this study, we propose to detect and prevent phishing infected e-mails. To achieve this, Feature Selection techniques and Neural Networks were combined, which allow to determine the probability that an email is of a Phishing type. As proof of concept, the algorithm was implemented on three datasets, which have been compiled during nine months from public email lists obtained from Debian, and run in order to be analyzed in a virtual environment. All emails were also compared against black Phishing lists obtained from PhishTank, in order to validate Phishing and HAM (non-Phishing emails). The main contribution of this study has been to design and implement a low-cost countermeasure, allowing to detect and mitigate phishing attacks, which have been already stored in the corporate e-mail server, using automatic learning methods.

The remainder of the article has been organized as follows: in Section II we discuss the structure of the used system, techniques and methods. In section III we validated and analyzed the obtained results with our proposal. Finally, section IV highlights the conclusions and future work lines.

II. METHODS

A. *Design Process*

According to (Trigas, 2012) [8], ”Scrum appears as a methodology aimed at technological products, which is based on the idea of creating short development cycles being called iterations or Sprints”. The Scrum methodology activities consist of a variety of different steps such as: (i) Selection of requirements; (ii) Task planning; (iii) Execution of the iteration (Sprint); (iv) Monthly meeting; (v) Daily meeting; Deliverable. The methodology establishes for each activity the personnel involved as well as the time and manner of execution. Hence, the prototype implementation is illustrated in Fig. 1.

1) Architecture diagram: As stated to [9], an architecture diagram helps to pose a complete view of the system to be built, as it indicates the structure and organization of the software components. In this sense, Fig. 1 exhibits the architecture proposed of the system, which allows browsing, reading, detecting, mitigating and alerting of Phishing threats as explained below. Upon receiving a new email on the server, it is sent to the mail client and subsequently processed in the MatLab software. In the software, its characteristics are extracted in order to be executed by the Feature Selection algorithm. After making the selection of characteristics, the learning vector is generated by means of the Neural Networks algorithm, thus determining whether it is Phishing or HAM (mail without Phishing). If it is determined that Phishing exists, the email is stored in a blacklist in a MySQL database.

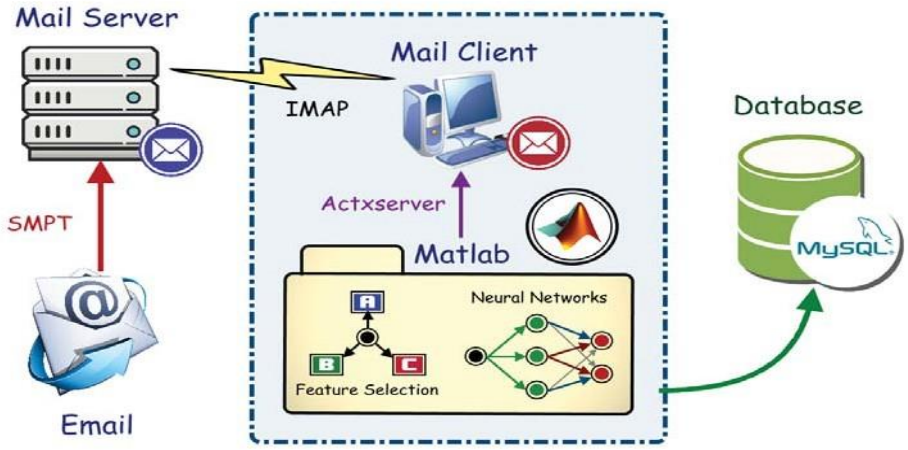

Fig. 1: Proposed architecture diagram

B. Development Process

Feature Selection method has been used, allowing the pre-processing of the characteristics of the emails and the elimination of unnecessary ones. In addition, Neural Networks were used for the construction of the machine learning vector. Fig. 2 illustrates the flow diagram of the given application.

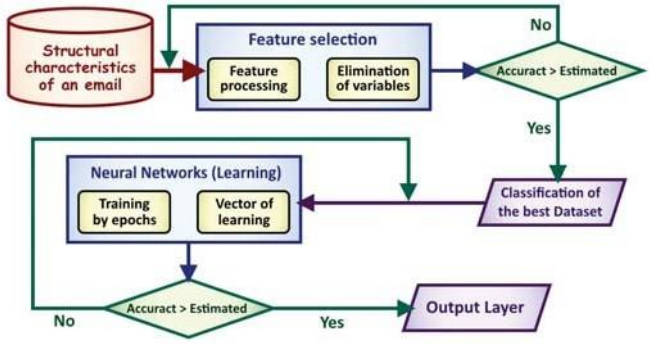

Fig. 2: Phishing detection and mitigation model flowchart

1) Feature selection Algorithm: In phishing, the attackers attempt to trick a potential victim in order to provide sensitive personal data involving fraudulent websites and emails that mimic legitimate ones. Within this context, Phishing attacks were generated, simulating the sending of emails according to the features in the email, which details are listed in Table I:

2) Neural Networks Algorithm: After extracting the important characteristics through Feature Selection, a reduced data set is generated from these features. With this data set, we proceeded to classify the data and to train the network. As we advanced to classify the data, we also defined the necessary inputs in order to generate the learning model by Neural Networks. These have been distributed by an equal amount of positive and negative examples.

TABLE I: Description of features

Features	Description
1. Mail has an image with an external link	This binary feature proposed by [10] represents the appearance of links in emails presenting images with the objective of detecting obfuscated URLs.
2. The sender of mail is not in the list of contacts	According to [11] the email is normal if the sender is a trusted contact.
3. The words in subject have been defined as unusual	This binary characteristic represents the appearance of words from a blacklist in the email. If the email contains the word from the blacklist, the email is abnormal and sets the value 1.
4. The email contains an attachment	To protect against these attacks, it is necessary to analyze the attached forms that contain suspicious field names.
5. The domain of the sender's email address contains more than 3 points	A large majority of current emails show URLs in dotted decimal format, which increases the suspicious factor
6. The mail contains a link disguised	Phishers disguise a destination website by hiding the URL. One method to hide the destination is to use the IP address of the website, instead of the host name.
7. The links contained in the email do not have an SSL certificate	It determines the links in the email that point to a website that encrypts the connection with an SSL digital certificate (Https).

3) Application combining Feature Selection and Neural Networks: After generating the predictive model, the software that would connect to the mail client was generated. In order to perform such task, the two algorithms have been combined, defining a main class.

4) Attack mitigation: After having detected a Phishing attack, there is a subsequent need to mitigate it. The mechanism implemented has been to revoke the user's access to the email detected as a threat and move it to a quarantine directory. The user receives an alert before executing the procedure. After the user is notified, the mail is moved to the quarantine directory. In this directory, all emails detected as threats are stored and may be identified according to their ID.

III. Evaluation of Results

A. Tests and performance analysis

For the validation of the obtained results, the source of information used was PhishTank's blacklist. Furthermore, the experimental configuration started with the acquiring of emails, which were collected from the website Mailing Lists Debian, 2019 by subscribing to the different lists they provide. These emails were hosted on a local server, obtaining a total of some three thousand emails. These emails were classified into three equivalent amounts of data sets, organized by time periods: (1) January to March; (2) April to June; (3) July to September.

The number of emails containing Phishing according to the classification made using the blacklist have been illustrated in Fig. 3 , where we evaluated the URLs recorded as Phishing. In this classification the percentage of Phishing in each data set is very low, since a total of 179 Phishing emails were obtained in the three time periods. This represents 5.96% of a total of 3000 analyzed emails.

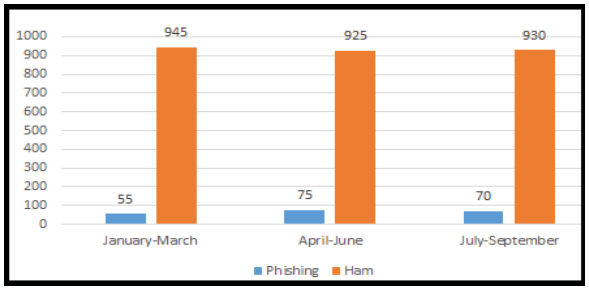

Fig. 3: Data set returned with the blacklist analysis

Figure 4 illustrates the results obtained after the execution of the proposed software. There is little variation from the previous results, given that a total of 204 Phishing emails have been encountered compared to the 179 emails returned by blacklist.

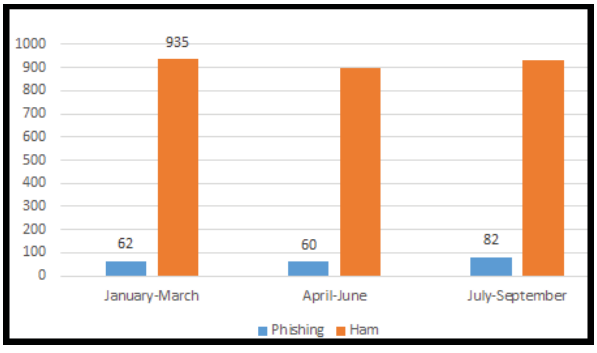

Fig. 4: Data set returned by the software proposed

Table II indicates the performance generated by the software, reaching some 92.9% of effectiveness with data set one, some 91.2% in the second and about 97.3% with the third. It is noted that the algorithm is capable of successfully detecting emails infected by Phishing with a maximum margin of error of around 8.8% and a minimum of some 2.7%.

TABLE II: Efficiency of the model

	Accuracy	Precision	Error	Recall
1	0.929	0.983	0.071	0.873
2	0.912	0.981	0.088	0.840
3	0.973	0.974	0.027	0.971

IV. Conclusions and Future Work

This study demonstrated the feasibility of detecting Phishing through the proper identification and use of the structural properties of an email, allowing a deeper study of the technical content that phishers use to perform illegal acts. The Agile Scrum methodology has been used during the implementation of the proposed software tool. Additionally, the Matlab process tool has been useful in the implementation of the automatic learning, feature selection and neural network algorithms. The obtained results from the concept tests are very promising, due to the fact that the implemented algorithms complement each other during detection. The evaluation of the results in the three data sets yielded an average accuracy of 93.9%. It also reveals the functionality of the proposed solution. As future work we have planned to generate a solution using Deep Learning and Bayesian Neural Networks.

Acknowledgments

The authors would like to thank the financial support of the Ecuadorian Corporation for the Development of Research and the Academy (RED CEDIA) in the development of this study, within the Project Grant GT-Cybersecurity.

References

[1] J. Mieres. (). Ataques informáticos. debilidades de seguridad comúnmente explotadas. 2009.
[2] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, ACM, 2007, pp. 60–69.
[3] S. Gastellier-Prevost, G. Granadillo, and M. Laurent, "Decisive heuristics to differentiate legitimate from phishing sites," Jun. 2011, pp. 1–9.
[4] A. Martin, N. B. Anutthamaa, M. Sathyavathy, M. M. S. Francois, and V. P. Venkatesan, "A framework for predicting phishing websites using neural networks," *CoRR*, vol. abs/1109.1074, 2011.
[5] M. Aburrous, M. Hossain, K. Dahal, and F. Thabtah, "Intelligent phishing detection system for e-banking using fuzzy data mining," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7913–7921, 2010, ISSN: 0957-4174.
[6] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proc. of the Anti-phishing Working Groups 2Nd Annual eCrime*, ser. eCrime '07, Pittsburgh, USA: ACM, 2007, pp. 60–69.
[7] A. Hamid, I. Rahmi, J. Abawajy, and T.-h. Kim, "Using feature selection and classification scheme for automating phishing email detection," *Studies in Informatics and Control*, vol. 22, pp. 61–70, Mar. 2013.
[8] M. Trigas. (). Metodología scrum. July 2012.
[9] Y. Li and S. Manoharan, "A performance comparison of sql and nosql databases," in *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, Aug. 2013, pp. 15–19.
[10] W. N. Gansterer and D. Pölz, "E-mail classification for phishing defense," in *Advances in Information Retrieval*, M. Boughanem, C. Berrut, J. Mothe, and C. Soule-Dupuy, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 449–460.
[11] F. Toolan and J. Carthy, "Feature selection for spam and phishing detection," in *2010 eCrime Researchers Summit*, Oct. 2010, pp. 1–12.