Detecting and Blocking Phishing URL using Excessive Machine Learning and the Concept of Artificial Intelligence

Asha Choudhary[1], Rakesh Rathi[2], Ankit Mundra[3]
[1]*Student, Department of CS & IT, Govt. Engg. College, Ajmer, India*
[2]*Professor, Department of CS & IT, Govt. Engg. College, Ajmer, India*
[3]*Assistant Professor, Department of CS & IT, Manipal University, Jaipur, India*

***Abstract*: Phishing URL is a major issue these days as the digitization is becoming more and more common and is the need for the future as well. Making things run online helps in saving a lot of time and also resources. Detection of phishing URL is tricky as there are various parameters which are needed to be considered before declaring a URL to be a phishing one, declaring a legitimate URL to be a phishing can cause a loss a lot of information and hence can create a lot of problem as well. The parameters over which URL credentials depends are discussed in this paper and the URL is tested over them and generated the appropriate results as well.**

***Keywords*: URL, Phishing, Classifier, MATLAB, Legitimate, Majority Methodology.**

1. Introduction

This paper deals with the importance of the URLs and about their security. A URL, short for universal resource locator, includes the protocol (ex. HTTP, FTP), the domain name (or IP address), and additional path information (folder/file). On the Web, a URL may address a Web page file, image file, or any other file supported by the HTTP protocol. URLs are often important to marketers in that they are part of the phrase “Add URL”, the process of submitting a site or page to another site, usually a search engine or directory.

A domain name is your website name. A domain name is the address where Internet users can access your website. A domain name is used for finding and identifying computers on the Internet. Computers use IP addresses, which are a series of number. However, it is difficult for humans to remember strings of numbers. Because of this, domain names were developed and used to identify entities on the Internet rather than using IP addresses.

A domain name can be any combination of letters and numbers, and it can be used in combination of the various domain name extensions, such as .com, .net and more.

A phishing website (sometimes called a "spoofed" site) tries to steal your account password or other confidential information by tricking you into believing you're on a legitimate website. You could even land on a phishing site by mistyping a URL (web address).

A legitimate URL is a URL which doesn’t steals any confidential information from your browser and never injects any virus into your system which may result into data sneaking or may crash down the whole system and may also corrupts the whole files and file system of it.

In this paper results are postulated which are generated by the software MATLAB.

MATLAB: Matrix Laboratory is a tool which work over the array data type and hence converts everything into a matrix and then process it accordingly.

On 30 function system will return result in the form of {-1, 0, 1} where -1 represents a phishing URL, 0 represents suspicious URL and 1 represents a legitimate URL.

2. Literature review

Phishing, one form of cyber-attacks, continues to be a growing concern not only to cyber security specialists but also to e-business users and owners. The severity of such cyber-attack vector is continuously growing with the exponential increase in digital information generation and the increased reliance of people and business on cyber space. The Anti-Phishing Working Group (APWG) has seen rapid growth in the number of unique phishing websites detected from 2014 to 2016 [19].

The average annual growth rate is 97.36% and is expected to continue to grow. Estimates of annual direct financial loss to the US economy caused by phishing activities range from $61 million to $3 billion.

To mitigate the increasing damage caused by phishing, a broad range of anti-phishing mechanisms have been proposed over the past two decades. These anti-phishing techniques can be categorized into three broad groups [12]: (1) Detective solutions (e.g., website filtering); (2) Preventive solutions (e.g., strong authentication and (3) Corrective solutions (e.g., Site takedown)

In this paper, we focus on detective solutions. More specifically, we look at software-based phishing detection schemes that are specialized in identifying and classifying phishing websites. This class of approaches is arguably more important than other approaches because it helps in reducing human errors. Preventative and corrective solutions take a different approach, but if the user behind the keyboard has been successfully tricked by a phishing attempt, and willingly submitted sensitive information, then no firewall, encryption software, certificates, or authentication mechanism can help in preventing the attack from materializing.

Software-based phishing detection also delivers improved results compared to detection by user education because phishing attacks normally aim at exploiting human weaknesses. For example, a study of phishing detection using user education shows a 29% false negative rate (FNR) for the best performance, while the software based approaches that are surveyed by the same study have FNR in the range of 0.1% to 10%. For this reason, we focus our study on software based phishing detection systems, and the term “phishing detection” will refer only to this form of detection in the rest of the paper.

Although the research area of phishing detection and classification is relatively rich, there is a lack of systematic analysis of the requirements, the capabilities, and the shortcomings of the existing anti-phishing techniques. For example, websites that offer identification and classification of phishing as a service have been popular in recent years, however, those services leverage different evaluation datasets from various sources at different time periods to validate their outcomes. Albeit those schemes may have similar performance results (e.g., in terms of false positive rate, true positive rate, etc.), it is difficult to compare their performance because of the variation in the evaluation datasets employed. Consequently, a systematic assessment of the datasets used to validate phishing detection approaches is desired, as well as necessary, in order to provide a foundation for comprehensive comparisons among different phishing detection schemes, and ultimately, select the best in practice.

In this study, we complement the existing survey papers on phishing detection, by providing a broad systematic analysis of software based antiphishing approaches. The authors focus on studying, analyzing, and classifying the most significant and novel detection techniques, and pointed out the advantages and disadvantages of each approach. On the other hand, we present a more comprehensive systematic review of phishing detection schemes, not only from the perspective of detection algorithms, but also from a broader perspective that covers other important aspects including the phishing detection life cycle, taxonomy of phishing detection schemes, evaluation datasets, detection features, and evaluation metrics and strategies. The work focuses more on the attack side of phishing. More specifically, it presents details about phishing attacks including the anatomy of such attacks, why people fall in phishing attacks and how bad phishing is. However, it only provides a high level analysis of the state-of-the-art phishing countermeasures. In order to provide a systematic review of the phishing detection research, we first present the necessary information about the phishing attacks by answering three questions: (1) What is phishing?, (2) How does phishing work? and (3) What is the current status of phishing? Then, we conduct systematic review of phishing detection schemes in a detailed and comprehensive manner. Finally, Khonji et al. present a literature survey about anti-phishing solutions (e.g., user training, email filtering and website detection, etc.), including their classification, detection techniques and evaluation metrics. Compared to, we focus on the software based phishing website detection schemes, which are proved to be the most effective anti-phishing solutions and are not systematically studied.

3. Proposed work

The implemented algorithm works on feature extraction based process, the URL has been asked as input for testing its credibility. The URL then undergoes through the functions which tests the URLs feature which are discussed in the upcoming sections.

The features that have been used for the information extraction are divided into four major categories a.) URL address Matrices, b.) URL Encoding Features, c.) URL Scripts Inclusion based Features and d.) URL Parameters based Features.

a) URL Address Matrices: This category deals with the URL length, IP address inclusion, tiny URL services, symbols inclusions, https current age parameters, URL age parameter, Fav icon, URL ports and accessibility of the https protocol.

b) URL Encoding Features: MATLAB can read the complete URL from a web browser and can display the HTML code along with title, head and body. Parameters which comes under this category are URL requests, Anchor URLs, excessive use of mail function or redirecting mails to a different domain name is also a sign of a phishing URL.

c) URL Scripts Inclusion: This function category check the URL encode on Forwarding links, Status Bar information, Right Click Disability, Pop Ups and Iframe presence, though these features are sometimes do not indicate the integrity of the URL as the banking websites keeps right click disable on their website.

d) URL Parameters: This category of function checks the URL over the registration and web performance parameters. Features like age of the domain in terms of years to come, DNS address verification, website traffic matrices, Page Rank Matrices, Google Website Index and Host IP Address verifications are done.

These feature extraction is the base for any URL detection algorithm. Here in this thesis as well it serves as a first check pass for any user entered URL or for any fetched URL.

Dynamic Database: In this algorithm the database is keep getting updated on every iteration and it helps the algorithm to get improved as well. The technical explanation of it can be stated as on production code iteration for single simulation the entered URL will be sent to the database list along with its parameters and the end result value of {-1, 0, 1}. From the feature function test the URL will get the score set which will decide the integrity of the URL whether it's a phishing URL or a legitimate one. Though over the sandbox code simulation the system will not update the database as it uses the same database for the testing.

Existing Database Checks and Validation: In this algorithm another database is referred as a legitimate repository database where the legitimate URLs are stored along with all their parameters and after the Feature Function's test the algorithm will refer to the repository database and then the score for legitimate URL or for phishing URL will get settle. Existing database ensures that the established domain name like amazon, Facebook, BushyThings and many more will always get the legitimate status and is never reported as a phishing URL due to reasons like absence of favicon as these features are becoming obsolete in order to make website load faster.

Score Updates and Check Average Score Calculations: A method has been introduced in this algorithm to make it more advance and up to the date w.r.t. the industry norms, the website may get a legitimate status even after getting a negative rating from the feature function test but due to the other parameters which are being involved. So there is a need of updating and calculating average scores which will then be taken into account while declaring a URL to be a legitimate one or the phishing one.

Add to Records: All the testing URLs which are being simulated under the production code runtime environment will be added to the existing database as per the acceptance of that particular database table.

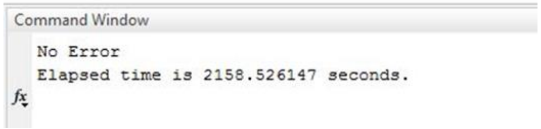
The designed algorithm works over the excessive model of machine learning, it not only analyses the previous database but also keeps them updated along with that it focuses over many other parameters which are externally affecting any URL and its integrity, MATLAB tools helps in reading the web encodes which are of high importance after considering the fact that the feature function test parameters requires web encodes in order to find and examine the internal links and scripts as well. This enables algorithm AI's domain as well and can be developed as an AI based detection model which will be able to make serious decision on its own.

4. Results

This algorithm has been tested over the dataset of 11000 URLs and the simulation time has also been recorded for 500 URLs as the System takes a lot of time to create database and then to check it on iterations is also a daunting task for the systems.

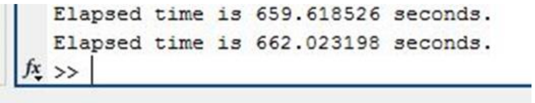
Fig. 1 shows the time taken by the MATLAB working environment to generate the database of 500 URLs.

From the database one by one every URL will be tested over the function and hence generates a result as per the set instructions.



```
No Error
Elapsed time is 2158.526147 seconds.
```

Fig. 1. Time Taken by the system to generate the database



```
Elapsed time is 659.618526 seconds.
Elapsed time is 662.023198 seconds.
fx >>
```

Fig. 2. Time taken by the algorithm to generate the result sheets

System will take priority coefficients into the account and will calibrate the results accordingly and noted it down into the excel sheet.

5. Conclusion

The above proposed work and the results we got after the simulation suggests that the regular database will be required to maintain to block and track down the history of the phishing websites and the features those have been implemented must get proper feedback so that they can improve themselves as well.

References

[1] Anti-Phishing Working Group et al. APWG Phishing trends reports, 2010 - 2016.

[2] Jason Hong. The state of phishing attacks. Communications of the ACM, 55(1):74–81, 2012.

[3] Saeed Abu-Nimeh, Dario Nappa, Xinlei Wang, and Suku Nair. A comparison of machine learning techniques for phishing detection. In Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit, pages 60–69. ACM, 2007.

[4] Issa Khalil, Saurabh Bagchi, and Ness Shroff. Analysis and evaluation of secos, a protocol for energy efficient and secure communication in sensor networks. Ad Hoc Networks, 5(3):360–391, 2007.

[5] Rajesh Krishna Panta, Saurabh Bagchi, and Issa M Khalil. Efficient wireless reprogramming through reduced bandwidth usage and opportunistic sleeping. Ad Hoc Networks, 7(1):42–62, 2009.

[6] Zuochao Dou, Issa Khalil, Abdallah Khreishah, and Ala Al-Fuqaha. Robust insider attacks countermeasure for hadoop: Design and implementation. IEEE Systems Journal, 2017.

[7] Zuochao Dou, Issa Khalil, and Abdallah Khreishah. CLAS: A novel communications latency based authentication scheme. Security and Communication Networks, 2017.

[8] Zuochao Dou, Issa Khalil, and Abdallah Khreishah. A novel and robust authentication factor based on network communications latency. IEEE Systems Journal, 2017.

[9] Ammar Gharaibeh, Mohammad A Salahuddin, Sayed J Hussini, Abdallah Khreishah, Issa Khalil, Mohsen Guizani, and Ala Al-Fuqaha. Smart cities: A survey on data management, security and enabling technologies. IEEE Communications Surveys & Tutorials, 2017.

[10] Issa Khalil and Saurabh Bagchi. Secos: key management for scalable and energy efficient crypto on sensors. Proceedings of IEEE Dependable Systems and Networks (DSN), 2003.

[11] Issa Khalil, Ting Yu, and Bei Guan. Discovering malicious domains through passive dns data graph analysis. In Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, pages 663–674. ACM, 2016.

[12] Issa Khalil, Ismail Hababeh, and Abdallah Khreishah. Secure inter cloud data migration. In Information and Communication Systems (ICICS), 2016 7th International Conference on, pages 62–67. IEEE, 2016.

[13] Ponnurangam Kumaraguru, Yong Rhee, Steve Sheng, Sharique Hasan, Alessandro Acquisti, Lorrie Faith Cranor, and Jason Hong. Getting users to pay attention to anti-phishing education: evaluation of retention and

transfer. In Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit, pages 70–81. ACM, 2007.

[14] Ponnurangam Kumaraguru, Steve Sheng, Alessandro Acquisti, Lorrie Faith Cranor, and Jason Hong. Teaching johnny not to fall for phish. ACM Transactions on Internet Technology (TOIT), 10(2):7, 2010.

[15] Steve Sheng, Bryant Magnien, Ponnurangam Kumaraguru, Alessandro Acquisti, Lorrie Faith Cranor, Jason Hong, and Elizabeth Nunge. Antiphishing phil: the design and evaluation of a game that teaches people not to fall for phish. In Proceedings of the 3rd symposium on Usable privacy and security, pages 88–99. ACM, 2007.

[16] Mahmoud Khonji, Youssef Iraqi, and Andrew Jones. Phishing detection: a literature survey. IEEE Communications Surveys & Tutorials, 15(4):2091–2121, 2013.

[17] Steve Sheng, Mandy Holbrook, Ponnurangam Kumaraguru, Lorrie Faith Cranor, and Julie Downs. Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 373–382. ACM, 2010.

[18] Mahmoud Khonji, Youssef Iraqi, and Andrew Jones. Phishing detection: a literature survey. IEEE Communications Surveys & Tutorials, 15(4):2091–2121, 2013.

[19] Gaurav Varshney, Manoj Misra, and Pradeep K Atrey. A survey and classification of web phishing detection schemes. Security and Communication Networks, 2016.