Phishing URL Detection Using Decision Tree

1st Abhilasha Jayaswal
College of Engineering
Drexel University
Philadelphia, USA
aj842@drexel.edu

2nd Mruga Shah
College of Computing and Informatics
Drexel University
Philadelphia, USA
mts339@drexel.edu

***Abstract*—Computer Science leads lots of activities in many fields in today's era leading to an increase in the number of cyber activities. The rise in use of Internet has caused an increase in the number of computer based fraudulent and malicious cyber-attacks. Most serious and relentless of these cyber-attacks is Phishing. Phishing is a cyber attack that attempts to steal users sensitive data such as passwords, credit card details.It has affected and continues to affect all the major industrial and commercial sectors, causing a lot of user data to be misused. To minimize loss of crucial and financial data it is important to detect phishing URL from the legitimate ones. The proposed approach incorporates analyzing a data-set of 10000 legitimate and 10000 phishing URLs using ID3 Decision Tree using lexical and host based features.**

***Index Terms*—Phishing, URL, Decision Tree, Legitimate URL, IP Address, Host Name, Cyber Attack.**

I. Background

For their mundane work and tasks people have started depending more upon technology, in today's world.In a world driven by cloud computing, online transactions, social network and automated processes, technology is constantly evolving. Cyber criminals are also constantly progressing with this evolution in technology.An unparalleled escalation in cyber-crime is witnessed by businesses spanning across many domains and applications.In fact, in todays age it is almost vital to have an online presence to make a venture successful. These cyber attacks range from Denial of Service to Man in the Middle, and from Malwares to Spams, Phishing to Spoofing. Out of all these attacks phishing is considered

The global address of documents and resources on the internet is known as Uniform Resource Locator and is abbreviated as URL. URL is compromised of three parts, first the protocol identifier, second the resource name (domain/host name or IP address) and third the path or parameters. A colon and two forward slashes are used to separate the protocol identifier from the resource name. Phishing or malicious URLs are the ones that are used to launch either a phishing attack or other cyber attacks. Some popular URL based cyber-attacks are social engineering, spam, phishing and drive-by download [1].

Phishing refers to the cyber attack when one website pretends to be a legitimate website there by trying to sell bogus products or tricking the users to reveal sensitive information, eventually leading to identity or money theft. In general a mail or text message is used to send a well drafted message like your account is about to expire, or so and so offer is added to your account accompanying the phishing or malicious URL luring the users to open the phishing URL and enter their credentials.

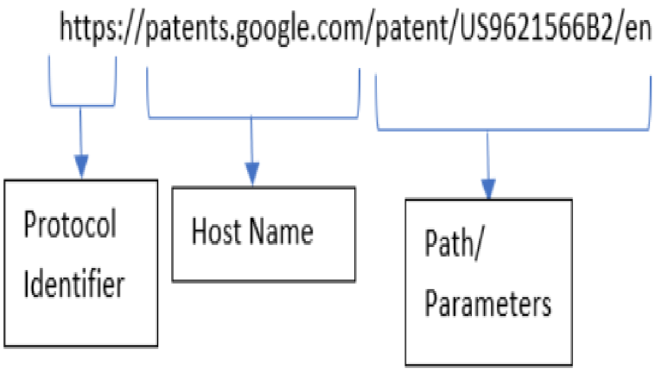

Fig. 1. Parts of URL.

Phishing has continued to be the topmost threat vector amongst cyber criminals because it is difficult to compromise the security system and firewall of a computer system but it is way easier to trick the user to click a malicious URL making it appear to be same as any legitimate URL, thus exploiting human vulnerabilities [2]. Both companies and end users are targeted by the attackers by sending phishing emails.

According to the Phishing Trends and Intelligence(PTI) Report 2018 threat attackers have become more enterprise focused rather than focusing on consumers. They target enterprises by pretending to service that most of the businesses rely on like email service providers, software service providers, etc. According to the report more than 12000 phishing attacks are investigated every month and a tremendous growth of 273% is seen in attacks against Software as a Service (SaaS) providers. The number of phishing domains hosted on HTTPS domain have grown to almost 33% in the last quarter of 2017 from less than 5% in 2016 [3].

Cyber attackers also use use Cybersquatting and Typosquatting for launching phishing attacks. Cybersquatting refers to using domain name that have registered name with the intent of taking advantage of a trademark or a brand-name belonging to someone else. Typosquatting is a type of cybersquatting that target users who incorrectly type the website address.

Attackers register fake websites that imitate the actual website and whenever the user makes any typographical error they may be led to an alternative website owned by the attacker [2].

II. Related Work

There is ample related work that is done till date on phishing detection and web security. Researchers have used both non-machine learning and machine learning techniques to identify phishing URLs.

Sujata, Niels, Monica and Aviel in their paper have discussed a way of distinguishing a phishing URL from the benign ones using several features. A logistic regression filter is modeled using these features. A several million URLs are classified using this model. The model produces high efficiency and accuracy. Around 777 new phishing URLs were found everyday and 8.24% of the users who viewed phishing URLs were potential victims. Features used by the authors were page rank, presence of the URL in the index, scores to quantify the quality of the page, presence of URL in the White Domain Table, type of URL, type 1-host obfuscated, type 2-,and presence of certain word tokens [9].

The authors of [10] have compared the performance of 9 different machine learning algorithms for detection of phishing websites. This 9 algorithms includes Support Vector Machines, Random Forest, Naive Bayes, Bayesian Additive Regression Trees, AdaBoost, Classification and Regression Trees, Logistic Regression and Bagging. The dataset consisted of 1500 legitimate and 1500 URL sites. AdaBoost performed the best amongst all the machine learning techniques used and 7 out of 9 machine learning techniques performed better than the traditional method used for classification.

Fergus and Joey in their paper have used a small dataset and classifier ensembles to examine the task of automated phishing detection. A number of classification algorithms were evaluated and the classifier ensembles was created using the best classifiers obtained through evaluation. In terms of accuracy, precision and F_1 score the ensembles performed no better than the individual algorithms. However, in terms of recall metric the ensembles performed much better than the individual techniques. In order to incorporate the recall improving feature with the good precision obtained for the best classification algorithms C5.0, R-Boost method was introduced [11].

Xun, John and Jeremy analyzed users behaviours in order to detect phishing websites. They have discussed how they have design and implemented this method so as to make it hard to circumvent and shown that it is an accurate method, and also have discussed about the methods' unique strength to protecting users from phishing threats. User-Behaviour Based Phishing Detection System (UBPD) is not developed with the intent to replace existing techniques. Rather it should be used to complement other techniques, to provide better overall protection. UBPD has three components the user profile, the monitor and the detection engine. UBPD has two modes: the training mode and the detection mode. In training mode the user updates its profile manually or the user profile is automatically updated with the currently unknown binding relationship by an automatic method. The UBPD calculates the phishing score in detection mode to decide whether the current web-page is phishing or not [12].

In [13], the authors have used real-world phishing data sets that have 177 initial features to evaluate correlation-based and wrapper-based feature selection techniques for phishing detection. The dataset used contains more than 16000 phishing URLs and more than 32000 non-phishing URLs. Two feature space searching techniques: genetic selection and greedy forward selection techniques are used to compare these feature selection techniques. Applying an effective feature selection procedure for experiments generally results in improvements that are significant statistically in the classification accuracies of among others Nave Bayes, Logistic Regression and Random Forests, in addition to improved efficiency in training time.

The authors have provided a research survey conducted on classification techniques for detecting phishing URLs bye various researchers. 4,500 URLs and several classification algorithms were used to perform the experiments. The tree-based classifiers provide maximum accuracy according to the observed results. The proposed system just analyzes the URL structure to recognize phishing URLs. The phishing sites are neither clicked nor entered by the authors. Hence, the time taken to analyze is drastically because the authors did not have a look at the URL content or page details. The dataset used consisted of 2500 genuine URLs, and 2000 phishing URLs. The features selected for the purpose of classification were Lexical Features: presence of IP address, count of number of dots present and presence of unknown known, URL-Based Features: presence of security sensitive words or suspicious symbols and misplaced top level domain, Network- Based Features: no. of sites linked to the URL and received traffic, and Domain- Based Features: domain age [14].

III. Proposed Work

In our proposed system, we identify URLs as phishing or legitimate by analyzing the features such as lexical features, host-based features that are specific to either phishing URLs or legitimate URLs. Detection of phishing URLs can be described as a classification problem. For the purpose of classification, the data needed is required to be labeled and should have samples as phishing or legitimate for the training phase. The dataset used should be precise that is the ones labeled as phishing should be absolutely identified as phishing, likewise the legitimate URLs should also be absolutely identified as legitimate.

A. *Data Set*

For performing this classification a dataset compromising of 20000 URLs was collected. This data comprises of 10000 phishing URLs and 10000 legitimate URLs. The data has been collected from the following different sources:

1) Legitimate URLs: The legitimate URLs dataset is collected from the open source dataset provide by Amazon as Alexa.com. The dataset contains a list of top one million legitimate websites. The metric for Alexa dataset is based on http requests from the users. Any site is ranked according to the measure of unique visitors and number of page views. For the purpose of this project we have used the top 10000 legitimate website URLs. The csv file for the data can be downloaded from [7].

2) Phishing URLs: The phishing URLs dataset is collected from the open source dataset known as PhishTank. PhishTank is a community where anyone can report, verify or track phishing data for free. PhishTank is operated by OpenDNS, a company dedicated to making Internet safer through faster, smarter and more secure DNS. For the purpose of this project, we have used the top 10000 phishing website URLs. The csv file for the data can be downloaded from [8].

B. ***Feature Extraction***

A malicious URL and a legitimate URL can be differentiated on the basis of various features extracted from the URL. Besides URL-based features, there are other features like Domain-Based Features, Page-Based Features and Content-Based Features that can be used. URL-based features include length of URL, digit count in URL, number of dots in the URL, number of sub-domains in the URL and many more. Some useful Domain-based features include checking if the domain name is valid or is in the blacklist, number of days that passed since domain was registered and whether the registrant name is hidden. Page-based features give us information on user activity on the page that is opened using that particular URL. These features include global page-rank, country page-rank, position at Alexa Top one million sites, average visit duration, estimated visits of the domain on weekly or monthly basis, count of reference from Social Networks to the given domain and more. Content-based features process information about the page title, text in the body, images on the page, hidden text in the body to decide whether to login to the website [2].

For the scope of this project, we use URL-based features, also commonly known as Lexical-based features. We extract the following features from the URLs:

IP Address:

Attackers use IP address in the malicious URLs (instead of host name) to hide the actual URL, to make it look legitimate and also to shorten the URLs length. Legitimate URLs sometimes use IP addresses but only for internal networks and these are not available publicly [4]. We have used this as a factor to distinguish between the URLs.

Number of ".com"s in the URL:

Multiple occurrence of ".com" in the URL is suspicious [4]. This implies that there is a threat of phishing by using the method of redirecting request. For example, http://amazon.com/url?q=http://www.malicious.com will redirect a user to malicious.com who is trying to use amazon.com. Thus, if there is more than one ".com" in the URL, we classify it as malicious URL.

'@','%40' in the URL:

The presence of an '@' symbol causes a redirection attack. In the URL, anything before the '@' symbol is commented and everything after '@' is used to visit the page [5]. For example, 'http://google.com@http://malicious.com' will make the user visit 'http.malicious.com'. Also, ASCII encoding of '@' is '%40' thus, if in the above example, if we replace '@' with '%40', it will still redirect the user to 'http.malicious.com'. Hence, for classifying the URLs in our project, we check whether either of these symbols is present in the URL. If so, we classify them as Phishing URL.

Length of URL,host name:

On reading other papers, we found out that the length of the URL/host name are also key factors in classifying the URLs. From the study in [15] and [6], we find out that phishing URLs have a longer URL length than the legitimate URLs. Similarly, even host names for phishing URLs are longer than in the legitimate ones. Hence, while feature extraction from the URLs, we extract the length of the URL and the host name and use these lengths to classify the URLs.

Number of slashes/dots in the URL and host name:

Both Legitimate URLs and Phishing URLs have dots and slashes in the URLs and host name of the URL. Attackers cannot reduce the number of slashes in their URLs since have to attach the target domain/host name in the phishing URL as a deception. If the URL contains more than 5 slashes, it is considered to be Phishing URL [4]. Thus, in our feature set we include these features for URLs by referencing from the paper in [6]. From the paper we learn that if the URL contains more than 15 dots then it is a phishing URL and if its host name has more than 5 dots then it is a phishing URL.

These features are extracted and URLs are given class labels as 1- Legitimate URLs and as 0- Phishing URLs based on the extracted features and relative criteria using a python code that is developed on Linux (specifically Drexel University tux). In the python code, the csv file containing the list of 20000 URLs (10000 legitimate and 10000 phishing URLs) is read using the open() function. A loop is run through this list to extract features that are described above from each URL one by one. The URLs with their corresponding features are extracted into another csv file (with the help of "pandas" library in python). Also, another csv file is created where each feature is compared to a threshold assigned to it (will discuss in Section IV.A). On comparison, if the feature is closer to parameters of Legitimate URLs, then we replace the feature by the class label (1 for Legitimate URLs and 0 for Phishing URLs). This is done because, we need a specific decision for each feature while using our classifier(discussed in section

III.B) The source code for feature extraction is submitted along with the paper.

C. *Classifier Used*

In Machine Learning, Classification is a method that is used in identifying which set an observation belongs to. One example of classification is, if we have a dog, identifying whether it is a pug or a German shepherd based on the features(height, weight and others) of the dog. We can do binary classification or multi-class classification based on the number of classes we have [16]. There are various types of classifiers namely Support Vector Machines, Logistic Regression, Decision Tree Classifier, Naive Bayes, Neural Networks, Random Forest and more. Each of them are efficient in different situations. For the purpose of this project, we use ID3 Decision Tree as our classifier.

Decision tree is a type of supervised learning. In a decision tree, each internal tree node is the attribute, each branch descending from the internal node corresponds to one of the possible values of the attribute at that node, and each leaf node is the class label (or a decision) [17]. In a decision tree, we need to first identify which attribute will be the root node and what attributes will form the next nodes in the tree. For an ID3 decision tree, this is done by calculating information gain of each feature. The feature with the highest information gain(or lowest entropy) becomes the root node of the tree and the feature with the next highest information gain forms the node after the root node and so on [18].

The code for training and testing the decision tree is developed in MATLAB. The entropy/information gain are calculated each time we traverse a branch in the decision tree. The csv file containing feature classes is used as the data set that is first skewed to mix up the legitimate and phishing URL data and then first two/thirds of data is used to train the decision tree and remaining data is used to classify the URLs. On the basis of that, we calculate the true/false positives and negatives and check the accuracy with which the decision tree classifies the URLs. All of this is done using MATLAB code. The mathematical calculations are given in detail in the next section (III.D).

D. *Calculation involved in computation of Results*

Entropy is a measure that characterizes homogeneity of examples. The formula to calculate entropy is:

$$Entropy(S) = E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i \tag{1}$$

This formula gives us the entropy of S, relative to c-wise classifications and p_i is the proportion of S that belongs to class i.

Information Gain(IG) measures how much information does a feature give us about a class.

$$IG = E[parent] - (avg.weights) * (E[child]) \tag{2}$$

There are error types namely True positives(TP), True negatives(TN), False positives(FP), False negatives(FN) that we get when we use the test data and classify the URLs. In our example, we get a True positive when the legitimate URL is classified as Legitimate. We get True negative when Phishing URL is classified as malicious URL. We get False positive when a Phishing URL is classified as legitimate and we get False negative when a Legitimate URL is classified as malicious. The table I below explains it well:

TABLE I
ERROR TYPES

	Predicted positive	Predicted Negative
Positive Examples	**True Positives**	**False Negatives**
Negative Examples	**False Positives**	**True Negatives**

To evaluate the classification, we use several parameters namely Precision, Recall and Accuracy.

Precision is the percentage of things that were classified as positive and were actually positive. The formula is:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Recall is the percentage of true positives correctly identified.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

The formula to calculate the accuracy of the classifier is:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{5}$$

Depending on the thresholds we set for our classes, we get different values for the error types. By changing the threshold, we can create a PR curve that is called the precision-recall curve that visually tells us what threshold to choose for our classification.

IV. EVALUATIONS

A. *Dataset Information*

The csv file that contains the dataset, has the 20000 URLs and the URL features are extracted using the python code as explained in Section III.B. After extracting the features like length of URLs, number of dots/slashes and others as explained in section III.B, we use thresholds to make these features discrete.

Initially, the thresholds are decided using the paper given in [6]. These thresholds are given in the table II below:

TABLE II
THRESHOLDS BASED ON RESEARCH

Feature	**Legitimate URL**	**Phishing URL**
Length of the URL	$<$ 383 char	$>$ 383
No. of dots in URL	$<$ 16	$>$ 16
No. of slashes in URL	$<$ 6	$>$ 6
Length of the host name	$<$ 71 char	$>$ 71 char
No. of dots in host name	$<$ 6	$>$ 6

Using these thresholds, when we run the python code to assign discrete values to the features and later run the MATLAB code that has the decision tree, we get an accuracy of 0.973.

Next, we change the thresholds by observing the features of our data and using those thresholds to calculate accuracy. We decide the threshold by looking at the maximum value for each feature of legitimate URL. For example, if the maximum length of the legitimate URL from the 10000 legitimate URLs is 45, then we set the threshold to be 45 for that feature. Similarly, for other features. The table III for these features is given below:

TABLE III
THRESHOLDS BASED ON OBSERVING OUR DATASET

Feature	**Legitimate URL**	**Phishing URL**
Length of the URL	$<$ 46 char	$>$ 46
No. of dots in URL	$<$ 4	$>$ 4
No. of slashes in URL	$<$ 3	$>$ 3
Length of the host name	$<$ 37 char	$>$ 37 char
No. of dots in host name	$<$ 4	$>$ 4

On over-fitting the data, when we run the python code to assign discrete values to the features and later run the MATLAB code that has the decision tree, we get an accuracy of 0.9878

Later, we relax the thresholds a bit to avoid over fitting of data as seen in the table below.

TABLE IV
THRESHOLDS BASED ON RELAXING THE THRESHOLDS

Feature	**Legitimate URL**	**Phishing URL**
Length of the URL	$<$ 70 char	$>$ 70
No. of dots in URL	$<$ 10	$>$ 10
No. of slashes in URL	$<$ 4	$>$ 4
Length of the host name	$<$ 50 char	$>$ 50 char
No. of dots in host name	$<$ 5	$>$ 5

On relaxing the over-fit, when we run the python code to assign discrete values to the features and later run the MATLAB code that has the decision tree, we get an accuracy of 0.9952.

V. CONCLUSIONS

Phishing community poses a constant challenge for security analyst throughout the world as new and advanced techniques are developed everyday. Through this project, we demonstrate that phishing attacks can be avoided by using simple Machine Learning techniques like classification algorithms. We evaluated our proposed solution on real world databases of URLs to detect the malicious URLs. Results that we get by experiment, show that such techniques can detect the phishing URLs with an accuracy of 99.5%. This kind of accuracy also shows that using only Lexical-based features of the URL can provide a lot of information about the URL. Adding other non-lexical features can make us reach a point where we can ideally identify any URL that is malicious.

We also observe during our experiment that increasing the number of dataset for training and testing purposes improves the accuracy of the decision tree many-folds. Earlier, when we tried to run the decision tree using a smaller dataset of 7000 URLs(containing 5000 Legitimate URLs and 2000 Phishing URLs), we got an accuracy of around 80% for our classifier but when we increased our database of URLs containing 10000 Legitimate URLs and 10000 Phishing URLs, we were able to classify with 99.5% accuracy. The accuracy also changes on changing the thresholds for the features. Hence, we also draw another conclusion that decisions of threshold selection should also be carefully taken.

VI. FUTURE WORK

In future, we plan to use an extensive database that is a bigger list of legitimate and phishing URLs with additional features for each URL. More number of features for each URL will definitely improve our accuracy results. Features such as page rank, average page-views for the URL, average page visit duration, category of domain, is domain name blacklisted, is the registrants name hidden [2] and more, would provide more information about the URL and further improve our results by reducing the number of False positives and False negatives classification.

For the scope of this project, we used only ID3 Decision tree as our classifier. Although, there are a multitude of classifiers that could be used. Some examples of these classifiers are J48 Decision tree, Random Forest, Naive Bayes classifier, Support Vector Machines, Neural Networks and Logistic Regression. Our future work would be to compare the results obtained from each of those classifiers and consequently better classify the URLs.

Thus, extensions to this project include feature expansion and classifier evaluation for improving the results of classification and applying these machine learning techniques to the real world for avoiding phishing scams.

REFERENCES

[1] P. Shi, X. Yao, S. He, and B. Cui, Malicious URL detection with feature extraction based on machine learning, International Journal of High Performance Computing and Networking, vol. 12, no. 2, p. 166, 2018.

[2] E. Buber, Phishing URL Detection with ML, Towards Data Science, 08-Feb-2018. [Online]. Available: https://towardsdatascience.com/phishing-domain-detection-with-ml-5be9c99293e5. [Accessed: 18-Mar-2019].

[3] 2018 Phishing Trends and Intelligence Report, Phishlabs, 2018.

[4] P. Agarwal and D. Mangal, A Novel Approach for Phishing URLs Detection, International Journal of Science and Research (IJSR), vol. 5, no. 5.

[5] D. Ranganayakulu and C. C., Detecting Malicious URLs in E-mail An Implementation, AASRI Procedia, vol. 4, pp. 125131, 2013.

[6] R. B. Basnet, A. H. Sung, and Q. Liu, Learning To Detect Phishing Urls, International Journal of Research in Engineering and Technology, vol. 03, no. 06, 2014.

[7] Alexa.com. [Online]. Available: http://s3.amazonaws.com/alexa-static/top-1m.csv.zip.

[8] PhishTank. Available: https://www.phishtank.com/developer_info.php.

[9] S. Garera, N. Provos, M. Chew, and A. D. Rubin, A framework for detection and measurement of phishing attacks, Proceedings of the 2007 ACM workshop on Recurring malcode - WORM 07, 2007.

[10] Miyamoto D., Hazeyama H., Kadobayashi Y. (2009) An Evaluation of Machine Learning-Based Methods for Detection of Phishing Sites. In: Kppen M., Kasabov N., Coghill G. (eds) Advances in Neuro-Information Processing. ICONIP 2008. Lecture Notes in Computer Science, vol 5506. Springer, Berlin, Heidelberg.

[11] F. Toolan and J. Carthy, Phishing detection using classifier ensembles, 2009 eCrime Researchers Summit, 2009.

[12] X. Dong, J. A. Clark, and J. L. Jacob, Defending the weakest link: phishing websites detection by analysing user behaviours, Telecommunication Systems, vol. 45, no. 2-3, pp. 215226, Feb. 2010.

[13] R. B. Basnet, A. H. Sung, and Q. Liu, Feature Selection for Improved Phishing Detection, Advanced Research in Applied Artificial Intelligence Lecture Notes in Computer Science, pp. 252261, 2012.

[14] K. V. Pradeepthi and A. Kannan, Performance study of classification techniques for phishing URL detection, 2014 Sixth International Conference on Advanced Computing (ICoAC), 2014.

[15] Jeeva, S.C. Rajsingh, E.B. Hum. Cent. Comput. Inf. Sci. (2016) 6: 10. https://doi.org/10.1186/s13673-016-0064-3

[16] Getting started with Classification, GeeksforGeeks, 09-Feb-2018. [Online]. Available: https://www.geeksforgeeks.org/getting-started-with-classification/. [Accessed: 19-Mar-2019]

[17] G. Drakos and G. Drakos, Decision Trees Decoded: Part 1, Towards Data Science, 06-Aug-2018. [Online]. Available: https://towardsdatascience.com/decision-trees-decoded-part-1-23b45f69111c. [Accessed: 19-Mar-2019]

[18] Decision Tree Introduction with example, GeeksforGeeks, 14-Feb-2018. [Online]. Available: https://www.geeksforgeeks.org/decision-tree-introduction-example/. [Accessed: 19-Mar-2019]