Phishing URL Detection Using Python And Machine Learning.

ABSTRACT:

Phishing attack is a simplest way to obtain sensitive information from innocent users. Aim of the phishers is to acquire critical information like username, password and bank account details. Cyber security persons are now looking for trustworthy and steady detection techniques for phishing websites detection. This paper deals with machine learning technology for detection of phishing URLs by extracting and analyzing various features of legitimate and phishing URLs. Decision Tree, random forest and Support vector machine algorithms are used to detect phishing websites. Aim of the paper is to detect phishing URLs as well as narrow down to best machine learning algorithm by comparing accuracy rate, false positive and false negative rate of each algorithm

Keywords:

Phishing attack, Machine learning

1. INTRODUCTION:

Nowadays Phishing becomes a main area of concern for security researchers because it is not difficult to create the fake website which looks so close to legitimate website. Experts can identify fake websites but not all the users can identify the fake website and such users become the victim of phishing attack. Main aim of the attacker is to steal banks account credentials. In United States businesses, there is a loss of US$2billion per year because their clients become victim to phishing [1]. In 3rd Microsoft Computing Safer Index Report released in February 2014, it was estimated that the annual worldwide impact of phishing could be as high as $5 billion [2]. Phishing attacks are becoming successful because lack of user awareness. Since phishing attack exploits the weaknesses found in users, it is very difficult to mitigate them but it is very important to enhance phishing detection techniques. The general method to detect phishing websites by updating blacklisted URLs, Internet Protocol (IP) to the antivirus database which is also known as “blacklist" method. To evade blacklists attackers uses creative techniques to fool users by modifying the URL to appear legitimate via obfuscation and many other simple techniques including: fast-flux, in which proxies are automatically generated to host the web-page; algorithmic generation of new URLs; etc. Major drawback of this method is that, it cannot detect zero-hour phishing attack. Heuristic based detection which includes characteristics that are found to exist in phishing attacks in reality and can detect zero-hour phishing attack, but the characteristics are not guaranteed to always exist in such attacks and false positive rate in detection is very high [3]. To overcome the drawbacks of blacklist and heuristics based method, many security researchers now focused on machine learning techniques. Machine learning technology consists of a many algorithms which requires past data to make adecision or prediction on future data. Using this technique, algorithm will analyze various blacklisted and legitimate URLs and their features to accurately detect the phishing websites including zero- hour phishing websites

2. DATASET:

URLs of benign websites were collected from www.alexa.com and The URLs of phishing websites were collected from www.phishtank.com. The data set consists of total 36,711 URLs which include 17058 benign URLs and 19653 phishing URLs. Benign URLs are labelled as “0” and phishing URLs are labelled as “1”.

3. FEATURE EXTRACTION

We have implemented python program to extract features from URL. Below are the features that we have extracted for detection of phishing URLs.

1) Presence of IP address in URL:

If IP address present in URL then the feature is set to 1 else set to 0. Most of the benign sites do not use IP address as an URL to download a webpage. Use of IP address in URL indicates that attacker is trying to steal sensitive information.

2) Presence of @ symbol in URL:

If @ symbol present in URL then the feature is set to 1 else set to 0. Phishers add special symbol @ in the URL leads the browser to ignore everything preceding the “@” symbol and the real address often follows the “@” symbol [4].

3) Number of dots in Hostname:

Phishing URLs have many dots in URL. For example http://shop.fun.amazon.phishing.com, in this URL phishing.com is an actual domain name, whereas use of “amazon” word is to trick users to click on it. Average number of dots in benign URLs is 3. If the number of dots in URLs is more than 3 then the feature is set to 1 else to 0.

4) Prefix or Suffix separated by (-) to domain:

If domain name separated by dash (-) symbol then feature is set to 1 else to 0. The dash symbol is rarely used in legitimate URLs. Phishers add dash symbol (-) to the domain name so that users feel that they are dealing with a legitimate webpage. For example Actual site is http://www.onlineamazon.com but phisher can create another fake website like http://www.online-amazon.com to confuse the innocent users.

5) URL redirection:

If “//” present in URL path then feature is set to 1 else to 0. The existence of “//” within the URL path means that the user will be redirected to another website [4].

6) HTTPS token in URL:

If HTTPS token present in URL then the feature is set to 1 else to 0. Phishers may add the “HTTPS” token to the domain part of a URL in order to trick users. For example, http://https-www□paypal-it-mpp-home.soft-hair.com [4].

7) Information submission to Email:

Phisher might use “mail()” or “mailto:” functions to redirect the user’s information to his personal email[4]. If such functions are present in the URL then feature is set to 1 else to 0.

8) URL Shortening Services “TinyURL”:

TinyURL service allows phisher to hide long phishing URL by making it short. The goal is to redirect user to phishing websites. If the URL is crafted using shortening services (like bit.ly) then feature is set to 1 else 0

9) Length of Host name:

Average length of the benign URLs is found to be a 25, If URL's length is greater than 25 then the feature is set to 1 else to 0

10) Presence of sensitive words in URL:

Phishing sites use sensitive words in its URL so that users feel that they are dealing with a legitimate webpage. Below are the words that found in many phishing URLs:- 'confirm', 'account', 'banking', 'secure', 'ebyisapi', 'webscr', 'signin', 'mail', 'install', 'toolbar', 'backup', 'paypal', 'password', 'username', etc;

11) Number of slash in URL:

The number of slashes in benign URLs is found to be a 5; if number of slashes in URL is greater than 5 then the feature is set to 1 else to 0.

12) Presence of Unicode in URL:

Phishers can make a use of Unicode characters in URL to trick users to click on it. For example the domain "xn--80ak6aa92e.com" is equivalent to "apple.com". Visible URL to user is "apple.com" but after clicking on this URL, user will visit to "xn--80ak6aa92e.com" which is a phishing site.

13) Age of SSL Certificate:

The existence of HTTPS is very important in giving the impression of website legitimacy [4]. But minimum age of the SSL certificate of benign website is between 1 year to 2 year.

14) URL of Anchor:

We have extracted this feature by crawling the source code oh the URL. URL of the anchor is defined by tag. If the tag has a maximum number of hyperlinks which are from the other domain then the feature is set to 1 else to 0.

15) IFRAME:

We have extracted this feature by crawling the source code of the URL. This tag is used to add another web page into existing main webpage. Phishers can make use of the "iframe" tag and make it invisible i.e. without frame borders [4]. Since border of inserted webpage is invisible, user seems that the inserted web page is also the part of the main web page and can enter sensitive information.

16) Website Rank:

We extracted the rank of websites and compare it with the first One hundred thousand websites of Alexa database. If rank of the website is greater than 10,0000 then feature is set to 1 else to 0.

4. MACHINE LEARNING ALGORITHM:

Three machine learning classification model Decision Tree, Random forest and Support vector machine has been selected to detect phishing websites

4.1 Decision Tree Algorithm:

One of the most widely used algorithm in machine learning technology. Decision tree algorithm is easy to understand and also easy to implement. Decision tree begins its work by choosing best splitter from the available attributes for classification which is considered as a root of the tree. Algorithm continues to build tree until it finds the leaf node. Decision tree creates training model which is used to predict target value or class in tree representation each internal node of the tree belongs to attribute and each leaf node of the tree belongs to class label. In decision tree algorithm, gini index and information gain methods are used to calculate these nodes.

4.2 Random Forest Algorithm:

Random forest algorithm is one of the most powerful algorithms in machine learning technology and it is based on concept of decision tree algorithm. Random forest algorithm creates the forest with number of decision trees. High number of tree gives high detection accuracy. Creation of trees are based on bootstrap method. In bootstrap method features and samples of dataset are randomly selected with replacement to construct single tree. Among randomly selected features, random forest algorithm will choose best splitter for the classification and like decision tree algorithm; Random forest algorithm also uses gini index and information gain methods to find the best splitter. This process will get continue until random forest creates n number of trees. Each tree in forest predicts the target value and then algorithm will calculate the votes for each predicted target. Finally random forest algorithm considers high voted predicted target as a final prediction.

4.3 Support Vector Machine Algorithm:

Support vector machine is another powerful algorithm in machine learning technology. In support vector machine algorithm each data item is plotted as a point in n-dimensional space and support vector machine algorithm constructs separating line for classification of two classes, this separating line is well known as hyperplane. Support vector machine seeks for the closest points called as support vectors and once it finds the closest point it draws a line connecting to them. Support vector machine then construct separating line which bisects and perpendicular to the connecting line. In order to classify data perfectly the margin should be maximum. Here the margin is a distance between hyperplane and support vectors. In real scenario it is not possible to separate complex and non linear data, to solve this problem support vector machine uses kernel trick which transforms lower dimensional space to higher dimensional space

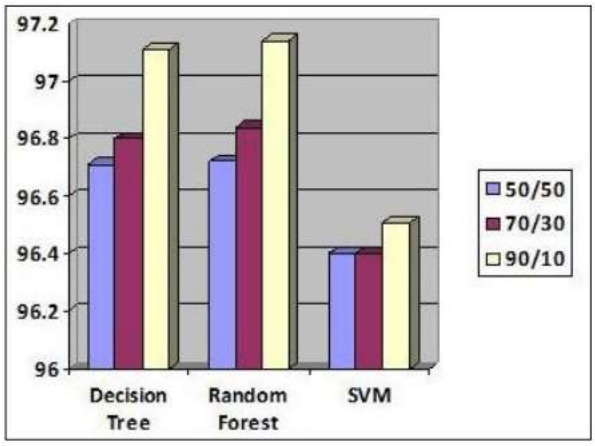

Fig. 1 Detection accuracy comparison

5. IMPLEMENTATION AND RESULT:

The scikit-learn tool has been used to import Machine learning algorithms. Dataset is divided into training set and a testing set in 50:50, 70:30 and 90:10 ratios respectively. Each classifier is trained using training set and testing set is used to evaluate performance of classifiers. The performance of classifiers has been evaluated by calculating classifier's accuracy score, false negative rate and false positive rate.

Table 1: Classifier's performance

Dataset Split ratio	Classifiers	Accuracy Score	False Negative Rate	False Positive Rate
50:50	Decision Tree	96.71	3.69	2.93
	Random Forest	96.72	3.69	2.91
	Support vector machine	96.40	5.26	2.08
70:30	Decision Tree	96.80	3.43	2.99
	Random Forest	96.84	3.35	2.98
	Support vector machine	96.40	5.13	2.17
90:10	Decision Tree	97.11	3.18	2.66
	Random Forest	**97.14**	**3.14**	2.61
	Support vector machine	96.51	4.73	2.34

The result shows that the Random forest algorithm1 m gives better detection accuracy which is 97.14 with lowest false negative rate than decision tree and support vector machine algorithms. The result also shows that the detection accuracy of phishing websites increases as more datasets are used as training datasets. All classifiers perform well when 90% of data is used as a training dataset. Fig. 1 show the detection accuracy of all classifiers when 50%, 70%, and 90% of data are used as training dataset and the graph clearly shows that detection accuracy increases when 90% of data is used as a training dataset and random forest detection accuracy is maximum than other two classifiers.

6. CONCLUSION:

This paper aims to enhance the detection method to detect phishing websites using machine learning technology. We achieved 97.14% detection accuracy using a random forest algorithm with the lowest false positive rate. Also, the result shows that classifiers give better performance when we used more data as training data. In the future hybrid technology will be implemented to detect phishing websites more accurately, for which random forest algorithm of machine learning technology and blacklist method will be used

7. REFERENCES:

[1] Gunter Ollmann, “The Phishing Guide Understanding & Preventing Phishing Attacks”, IBMInternet Security Systems, 2007.

[2] https://resources.infosecinstitute.com/category/enterprise /phishing/the-phishing-landscape/phishing-data-attack□statistics/#gref

[3] Mahmoud Khonji, Youssef Iraqi, "Phishing Detection: A Literature Survey IEEE, and Andrew Jones, 2013

[4] Mohammad R., Thabtah F. McCluskey L., (2015) Phishing websites dataset. Available: https://archive.ics.uci.edu/ml/datasets/Phishing+Websites Accessed January 2016

[5] http://dataaspirant.com/2017/01/30/how-decision-tree□algorithm-works/

[6] http://dataaspirant.com/2017/05/22/random-forest□algorithm-machine-learning/

[7] https://www.kdnuggets.com/2016/07/support-vector□machines-simple-explanation.html

[8] www.alexa.com

[9] www.phishtank.com