A Project report

on

Intelligent Rule Based Phishing Website Detection

Submitted By

Name of students (Roll No.)

Anjuru Lokesh	**-**	**18BCS006**
A.Sai Venkata Aditya	**-**	**18BCS011**
CH.N.V.Avinash	**-**	**18BCS021**
D.Geetha Krishna	**-**	**18BCS025**

Under the guidance of

UMA S

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

DHARWAD

Table of Contents

ABSTRACT :

This project aims to use an intelligent, versatile, and reliable method to recognise and classify e-banking and other phishing websites. In this project, we check if a given URL is Phishy or not by looking it up in a dataset, and if it's not there, we'll break it down based on factors like the URL, Domain Identity and Security, and encryption requirements, and determine whether it's Phishy or not. Phishing websites are those that ask users for personal information for a malicious reason.

OBJECTIVE:

Web phishing is one of many security threats to web services on the Internet. Web phishing aims to steal private information, such as usernames, passwords, and credit card details, by way of impersonating a legitimate entity. So this project mainly focuses on applying a machine learning framework to detect phishing websites.

1. INTRODUCTION

Phishing has become a major source of concern for security researchers in recent years because it is relatively simple to create a fake website that appears to be identical to a legitimate website. While experts can recognise fake websites, not all users can, and as a result, some users become victims of phishing attacks. The attacker's main goal is to steal bank account credentials. Because of a lack of user knowledge, phishing attacks are becoming more successful. Since phishing attacks take advantage of user flaws, it's difficult to minimise them, but it's critical to improve phishing detection techniques.

The "blacklist" approach is a general method for detecting phishing websites by adding blacklisted URLs and IP addresses to the antivirus database. To get around blacklists, attackers use clever techniques like obfuscation and many other simple techniques like fast-flux, where proxies are automatically created to host the web page; algorithmic generation of new URLs; and so on. The method's biggest flaw is that it can't detect zero-hour phishing attacks.

Heuristic-based detection, which uses characteristics that have been observed in real-world phishing attacks and can detect zero-hour phishing attacks, but the characteristics are not guaranteed to always be present in such attacks, and the false positive rate in detection is high.

Many security researchers are now focusing on machine learning techniques to solve the limitations of blacklist and heuristic-based methods. Machine learning technology is made up of a variety of algorithms that use historical data to make predictions about future data. The algorithm can evaluate various blacklisted and legitimate URLs and their features in order to accurately detect phishing websites, including zero-hour phishing websites, using this technique.

2. DATASET :

URLs of benign websites were collected from www.kaggle.com and The URLs of phishing websites were collected from www.phishtank.com. The data set consists of a total 2000 URLs which include 1000 benign URLs and 1000 phishing URLs. Benign URLs are labelled as “0” and phishing URLs are labelled as “1”.

3. FEATURE EXTRACTION:

We created a Python programme to extract features from URLs. The features extracted for detecting phishing URLs are listed below.

1) Presence of IP address in URL: If an IP address is present in the URL, the function is set to 1, otherwise it is set to 0. The majority of safe sites do not use an IP address as a URL to download a webpage. The use of an IP address in the URL means that the attacker is attempting to steal sensitive information.

2) Presence of @ symbol in URL: If the @ symbol appears in the URL, the function is set to 1, otherwise it is set to 0. When phishers use the special symbol @ in the URL, the browser ignores anything preceding the “@” symbol, and the real address frequently follows the “@” symbol

3) Number of dots in Hostname: Phishing URLs have a lot of dots in them. For example, http://shop.fun.amazon.phishing.com, where phishing.com is an actual domain name and the word “amazon” is used to trick users into clicking on it. In benign URLs, the average number of dots is three. If the number of dots in URLs exceeds three, the function is set to one; otherwise, it is set to zero.

4) Prefix or Suffix separated by (-) to domain: If the domain name is separated by a dash (-), the function is set to 1, otherwise it is set to 0. In legitimate URLs, the dash symbol is rarely used. Phishers add the dash symbol (-) to the domain name to give users the impression that they are dealing with a legitimate website. For example, the actual site is http://www.onlineamazon.com, but a phisher may create a fake website such as http://www.online-amazon.com to confuse unwary users.

5) URL redirection: If there is a "//" in the URL direction, the feature is set to 1; otherwise, it is set to 0. The presence of the character "//" inside the URL route indicates that the user will be redirected to another website.

6) HTTPS token in URL: If an HTTPS token is present in the URL, the function is set to 1, otherwise it is set to 0. To deceive users, phishers can append the "HTTPS" token to the domain portion of a URL. For example in the case, http://www.paypal-it-mpp-home.soft-hair.com

7) Information submission to Email: To redirect the user's information to his personal account, the phisher can use the "mail()" or "mailto:" functions. If such functions are present in the URL, the feature is set to 1, otherwise it is set to 0.

8) URL Shortening Services like "TinyURL": TinyURL service helps phishers to mask long phishing URLs by shortening them. The aim is to guide users to phishing websites. If the URL is created using a URL shortening service (such as bit.ly), function is set to 1, otherwise it is set to 0.

9) Length of Host name: The average length of safe URLs is discovered to be a 25. If the URL's length is greater than 25, the function is set to 1 otherwise to 0.

10) Presence of sensitive words in URL: Phishing websites use sensitive words in their URLs to make users believe they are visiting a legitimate website. The following words can be used in several phishing URLs: - 'confirm', 'account', 'banking','secure', 'ebayisapi', 'webscr', 'sign in', 'mail', 'instal', 'toolbar', 'backup', 'paypal', 'password', 'username', and so on;

11) Number of slash in URL: The number of slashes in benevolent URLs is found to be a 5; if the number of slashes in the URL is greater than 5, the feature is set to 1; if the number of slashes in the URL is less than 5, the feature is set to 0.

12) Age of SSL Certificate: The presence of HTTPS is critical in giving the impression that a website is legitimate. However, the SSL certificate of a benign website should be between 1 and 2 years old.

13) URL of Anchor: By crawling the source code of the URL, we were able to extract this function. The a> tag specifies the anchor's URL. The function is set to 1 if the a> tag has a maximum number of hyperlinks from another domain, otherwise to 0.

14) Website Rank: We took the rank of the websites and compared it to the first 100,000 websites in the Alexa database. If the website's rank is greater than 10,0000, the function is set to 1; otherwise, it is set to 0.

4.Random Forest Algorithm :

The random forest algorithm, which is built on the principle of the decision tree algorithm, is one of the most efficient algorithms in machine learning technology. The random forest algorithm generates a forest containing a large number of decision trees. A large number of trees results in a high level of detection precision.

The bootstrap method is used to build trees. To build a single tree, the bootstrap method selects features and samples from the dataset at random with replacement. Random forest algorithm, like decision tree algorithm, chooses the best splitter for classification from randomly chosen features. Random forest algorithm also uses gini index and knowledge gain methods to find the best splitter. This process will be repeated until the random forest has produced n trees.

Each tree in the forest predicts the target value, and the algorithm then calculates the votes for each target value predicted. Finally, the random forest algorithm uses the projected target with the most votes as a final prediction.

5. IMPLEMENTATION AND RESULT :

Machine learning algorithms were imported using the Scikit-learn platform. In 50:50, 70:30, and 90:10 ratios, the dataset is divided into training and testing sets. Each classifier is trained using a training set, and the output of the classifiers is evaluated using a testing set. The accuracy score, false negative rate, and false positive rate of classifiers were calculated to assess their efficiency.

6. INTEGRATED PROJECT WITH DATABASE :

We incorporated a MongoDB database into our project so that the URL is first checked against the database. If the URL cannot be located in the database, it is separated into processes and the ml algorithm is used to predict the outcome. The user is routed to the website if it is not phishy. If the URL is phyishy, it is immediately added to the database and the user is informed. It will locate the URL more reliably and effectively this way because it will not divide it further if it is found in the database.

7. WORKING OF PROJECT :

- We will have an interface build which will take an URL as an input.

- We will have a database in which all the Phishy websites found until now will be present.

- We will first check whether the input URL is present in the database or not.

- If the URL is present in the database then we will not redirect the user into that URL.

- And also we will display a message to the user that the particular URL is phishy.

- If the URL is not present in the database then we will divide the URL into 6 parts.

- Each part of the URL is checked through some steps and given some values as an output in each step.

- We had a ML model which was trained using some dataset

.

- This ML model will predict whether the website is phishy or not based on the values we got above.

- And if this ML model classifies the input URL as a phishy website then we will add the URL into our database automatically.

- We will not allow the user to redirect into the website if the URL is phishy and notifies the user with a notification.

- We will redirect the user into the website if it's not phishy.

8. TO TEST OUR PROJECT :

To try out our project, go to the link below, where you'll find all of the specifications and a readme that will guide you through the process.

Github Link

9. 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-learing/

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

[8] www.kaggle.com

[9] www.phishtank.com