**PROJECT DOCUMENTATION SUBMITTED TO TECHIONARY**



Phishing Site Detection

PROJECT DATE:

BATCH NO: 15

| TEAM MEMBERS | | | | |
| --- | --- | --- | --- | --- |
| S.No | NAME | ROLE | COLLEGE | CONTACT NO |
| 1. | Akanksha Agarwal | Web application Developer | Malla Reddy College of Engineering and Technology. | 7995576399 |
| 2. | Navya Dandu | Web Application Deployer | Malla Reddy College of Engineering and Technology. | 6302327830 |
| 3. | Vaishnavi BaluReddy | Machine Learning model Developer | Malla Reddy College of Engineering and Technology. | 7330885354 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PREPARED BY** | CSI-15 | **TITLE** | Phishing Site Detection | **DATE** | 17/8/2020 |
| **GUIDED BY** | Rupak Reddy | **TITLE** | Phishing Site Detection | **DATE** | 17/8/2020 |

Table of Contents

[PROBLEM STATEMENT 3](#_Toc18578293)

[PROJECT SCOPE 3](#_Toc18578294)

[PRODUCT OR SYSTEM FEATURES AND REQUIREMENTS 4](#_Toc18578295)

[PROJECT SUCCESS CRITERIA 4](#_Toc18578296)

[PROJECT SUCCESS CRITERIA 5](#_Toc18578297)

[1. STATEMENT OF WORK 6](#_Toc18578298)

[1.1 SCOPE OF WORK 6](#_Toc18578299)

[1.2 LOCATION OF WORK 6](#_Toc18578300)

[1.3 PERIOD OF PERFORMANCE 6](#_Toc18578301)

[1.4 SCHEDULE 6](#_Toc18578302)

[1.5 STANDARDS FOLLOWED 6](#_Toc18578303)

[1.6 ACCEPTANCE CRITERIA FOR USERS 6](#_Toc18578304)

[1.7 ADDITIONAL REQUIREMENTS 6](#_Toc18578305)

# PROBLEM STATEMENT

Problems faced by the users while using the current system

|  |
| --- |
| Phishing sites trick users by acting like legitimate sites and coerce them into giving out personal and sensitive information. It allows malicious users to access victims’ data and perform transactions on their account, steal information for fake identities, corrupt their computer systems or use their identities to infiltrate and steal high value organizational data from firms.  Even though phishing is over a decade old, many people are not familiar with how it works and still fall victim to this scam. |

# PROJECT SCOPE

Project Purpose and Justification

|  |
| --- |
| To avoid phishing attacks and losses due to scams, a phishing website detection software is developed.  We developed a full-stack web application using Django and deployed a Machine Learning model for Phishing site detection. Our application prompts the user to enter a site URL and detects whether the given URL is phished or not. We achieved an accuracy of 86% by using Support Vector Machine algorithm. We do predictions based on features like the total length of the URL, the number of subdomains in the URL, and the presence of an IP address. We incorporated a Python Virtual Environment and created the Model-View-Controller software. Our project aims at providing a simple, user-friendly web-based tool to detect malicious URLs in order to implement Cyber Security. |

# PRODUCT OR SYSTEM FEATURES AND REQUIREMENTS

|  |
| --- |
| The detection website consists of a designated space to enter a URL. The user only has to enter the URL and submit it which in turn gives them the response if it’s a phishing website or not.  For the application to be easily accessible by anybody, the user only needs to have an internet connection and a URL to be tested. The goal with such few requirements is make sure that everyone can conveniently use the product and stay risk free from phishing attacks. |

# PROJECT SUCCESS CRITERIA

|  |
| --- |
| For the application to detect phishing website, it must be self-sustainable without human intervention.  Hence, the project depends on a machine learning model with accuracy 85% or higher.  The datasets chosen to train the ML model should be authentic datasets.  The web application should avoid SQL injection, cross-site scripting, cross-site request forgery, and clickjacking. It should be stable and scalable and should incorporate a dynamic admin interface.  And the web application should be able to accept any URL and be able to give quick and accurate results.  We can say that all these criteria have been met successfully. |

# STATEMENT OF WORK

## SCOPE OF WORK

|  |
| --- |
| For detecting the URL of phishing websites, a machine learning model is first developed. For training the machine learning model, datasets of legitimate websites and spoofed websites are taken and feature selection is applied to these datasets. Since the features like the length of the URL and presence of IP address etc., are different for phishing and legitimate websites, after applying the feature selection the machine learning model is trained with Support Vector Machine (SVM) classifier. After saving the model with joblib.  Since Django is written in Python it makes it a great choice of web framework for deploying machine learning models. To set up a Django project we first need to install django and create a directory for the project with the name say Phishing.  Next, we create a project with django-admin startproject first\_project and then run python manage.py runserver and go to our local host. Next create web page called views.py in our project and add it to urls.py. To let Django know that we have this folder, add 'templates' to our 'DIRS' in settings.py.In views.py change what we're importing from Django from Django from HttpResponse and add home.html.To actually display an output add user\_input = requestt.GET["your url"] to our result function in views.py. When we enter an URL this will redirect to result.html to say whether the URL is phished or not.  While this particular project works as a website, it can be included in other web pages or as an extension when phishing websites needs to be detected. |

## LOCATION OF WORK

|  |
| --- |
| Online. |

## PERIOD OF PERFORMANCE

|  |
| --- |
| August 3, 2020 – August 21, 2020 |

## SCHEDULE

|  |  |  |  |
| --- | --- | --- | --- |
| **CATEGORY/TASK** | **WORK DONE** | **START DATE** | **END DATE** |
| Phase 1 - Planning |  |  |  |
| Phishing Sites and Web Framework research | Exploring research papers, existing projects and web application programming. | August 3, 2020 | August 5, 2020 |
| Roles designation | Picking Tasks | August 6, 2020 | August 6, 2020 |
| Phase 2 - Execution |  |  |  |
| Machine Learning Model – Dataset selection and Choosing algorithm | Exploring Datasets and picking the best algorithm based on accuracy | August 6, 2020 | August 7, 2020 |
| Web app- Deploying the machine learning model using python-based web framework Django | Used form tag to get the input as URL from the user, to run the ML model and display whether it is phished or not | August 7, 2020 | August 10, 2020 |
| Phase 3 - Monitoring |  |  |  |
| Machine Learning Model – Accuracy Improvement and Feature Selection | Improving the accuracy of the model and choosing the required features of the input URL | August 11, 2020 | August 13, 2020 |
| Styling- styling the web app is done by using CSS and Bootstrap | Padding, alignment, box, button are used | August 13, 2020 | August 16, 2020 |
| Phase 4 - Closing |  |  |  |
| Deploying and Testing the Website | Test the application with various URLs and fixing bugs. | August 16, 2020 | August 17,2020 |
| Preparing the Report and Presentation | Tying loose-ends and concluding the project | August 17, 2020 | August 20, 2020 |

## STANDARDS FOLLOWED

|  |
| --- |
| For the phishing website detection to work at the highest rate of accuracy, highly authentic datasets should be chosen for training the machine learning model. This is done by choosing phishing dataset from PhishTank and legitimate dataset from Alexa topsites. The machine learning model should also be of accuracy of 85% and above in order to give us the best result.  The phishing site detection web application is reassuringly secure, it avoids SQL injection, cross-site scripting, cross-site request forgery, and clickjacking. The web app is stable and scalable. It incorporates a dynamic admin interface. It is extremely user friendly and requires no prior cyber security knowledge for use.  The web application should be able to accept any URL and it should be able to give quick and accurate results. |

## ACCEPTANCE CRITERIA FOR USERS

|  |
| --- |
| While the input URL given to the web application is a string object, the machine learning model is trained on a dataset which is an object with rows and columns so a conversion is required.  For using the phishing website detection, the user need not have any previous knowledge of cyber security. It is aimed to help the users avoid getting scammed without requiring to learn and apply any new information. |

## ADDITIONAL REQUIREMENTS

|  |
| --- |
| We acquired our datasets from sources like Phishtank, Alexa and Kaggle which worked out well with our model and suited our project. Further use of elaborate datasets and advanced training can upscale the software performance leading to higher accuracy. |

CONCLUSION

At the end of the project we were able to achieve a dynamic, secure and scalable web application to prevent phishing attacks. Support Vector Machine, Machine Learning Algorithm was deployed on the web application built using the Django framework. We were able to develop a user-friendly software where the users can find out if a given website is phished or not with just a click of a button hence keeping them safe and preventing Cyber-Attacks.

REFERENCES

<https://archive.ics.uci.edu/ml/datasets/Phishing+Websites>

<https://medium.com/intel-software-innovators/detecting-phishing-websites-using-machine-learning-de723bf2f946>

<https://link.springer.com/article/10.1007/s12652-019-01311-4>

<https://towardsdatascience.com/phishing-domain-detection-with-ml-5be9c99293e5>

<https://anvil.works/learn/tutorials/jupyter-notebook-to-web-app>

<https://www.metacompliance.com/blog/5-ways-to-identify-a-phishing-website/>

<https://towardsdatascience.com/phishing-domain-detection-with-ml-5be9c99293e5>

<https://www.researchgate.net/publication/329836170_Malware_Analysis_and_Detection_Using_Reverse_Engineering_Technique>

<https://github.com/rshipp/awesome-malware-analysis>

<https://www.oreilly.com/library/view/machine-learning-and/9781491979891/ch04.html>

<https://zeltser.com/build-malware-analysis-toolkit/>

<https://www.mlq.ai/django-machine-learning/>

<https://www.djangoproject.com/start/overview/>

<https://www.irjet.net/archives/V7/i3/IRJET-V7I3115.pdf>