TEAM ID: PNT2022TMID40244

PROJECT NAME:WEB PHISHING DETECTION

NALAIYA THIRAN (IBM PROJECT)

TEAM MEMBERS

DURGA DEVI M
SINDHU J
SUBHIKSHA P
SURTHIKA G

Project Report Format

1. **INTRODUCTION**
   - 1.1 Project Overview
   - 1.2 Purpose
2. **LITERATURE SURVEY**
   - 2.1 Existing problem
   - 2.2 References
   - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   - 3.1 Empathy Map Canvas
   - 3.2 Ideation & Brainstorming
   - 3.3 Proposed Solution
   - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   - 4.1 Functional requirement
   - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   - 5.1 Data Flow Diagrams
   - 5.2 Solution & Technical Architecture
   - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   - 6.1 Sprint Planning & Estimation
   - 6.2 Sprint Delivery Schedule
   - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   - 7.1 Feature 1
   - 7.2 Feature 2
   - 7.3 Database Schema (if Applicable)
8. **TESTING**
   - 8.1 Test Cases
   - 8.2 User Acceptance Testing
9. **RESULTS**
   - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

    Source Code

    GitHub & Project Demo Link

Introduction

Project Overview:

Phishing is a social engineering attack that aims at exploiting the weakness found in system processes as caused by system users. For example, a system can be technically secure enough against password theft, however unaware end users may leak their passwords if an attacker asked them to update their passwords via a given Hypertext Transfer Protocol (HTTP) link, which ultimately threatens the overall security of the system or over, technical vulnerabilities (e.g. Domain Name System (DNS) cache poisoning) can be used by attackers to construct far more persuading socially-engineered messages (i.e. use of legitimate, but spoofed, domain names can be far more persuading than using different domain names). This makes phishing attacks a layered problem, and an effective mitigation would require addressing issues at the technical and human layers.

Since phishing attacks aim at exploiting weaknesses found in humans (i.e. system end-users), it is difficult to mitigate them. For example, as evaluated in end-users failed to detect 29% of phishing attacks even when trained with the best performing user awareness program . On the other hand, software phishing detection techniques are evaluated against bulk Phishing attacks, which makes their performance practically unknown with regards to targeted forms of phishing attacks. These limitations in phishing mitigation techniques have practically resulted in security breaches against several organizations including leading information security providers

In order to address the limitations of the previous definitions above, we consider phishing attacks as semantic attacks which use electronic communication channels (such as EMails, HTTP, SMS, VoIP, etc. . .) to communicate socially engineered messages to persuade victims to perform certain actions (without restricting the actions) for an attacker's benefit (without restricting the benefits)Phishing is a type of computer attack that communicates socially engineered messages to humans via electronic communication channels in order to persuade them to perform certain actions for the attacker's benefit. For example, the performed action (which the attacker persuades the victim to perform it) for a PayPal user is submitting his/her login credentials to a fake website that looks similar to PayPal. As a perquisite, this also implies that the attack should create a need for the end-user to perform such action, such as informing him that his/her account would be suspended unless he logs in to update certain pieces of information

Purpose:

Web Phishing Detection Category: Machine Learning Objective A phishing website is a common social engineering method that mimics trustful uniform resource locators (URLs) and webpages. The objective of this project is to train machine learning models on the dataset given to predict phishing websites**.**

There have been several recent studies against phishing based on the characteristics of a domain, such as website URLs, website content, incorporating both the website URLs and content, the source code of the website and the screenshot of the website . However, there is a lack of useful anti-phishing tools to detect malicious URL in an organization to protect its users. In the event of malicious code being implanted on the website, hackers may steal user information and install malware, which poses a serious risk to cybersecurity and user privacy. Malicious URLs on the Internet can be easily identified by analyzing it through Machine Learning (ML) technique

Phishing detection schemes which detect phishing on the server side are better than phishing prevention strategies and user training systems. These systems can be used either via a web browser on the client or through specific host-site software.presents the classification of Phishing detection approaches. Heuristic and ML based approach is based on supervised and unsupervised learning techniques. It requires features or labels for learning an environment to make a prediction. Proactive phishing URL detection is similar to ML approach. However, URLs are processed and support a system to predict a URL as a legitimate or malicious

LITERATURE SURVEY

Existing Problem:

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. Phishing may be a style of broad extortion that happens once a pernicious web site act sort of a real one memory that the last word objective to accumulate unstable info, as an example, passwords, account focal points, or MasterCard numbers. all the same, the means that there square measure some of contrary to phishing programming

- ✓ In the Existing problem the user directly put the URL and get the output in the same page.
- ✓ Do not explain the project details and Add URL options
- ✓ In the Existing system was not accurately Provide the output on the web site is phishing or not.
- ✓ In the Existing system do not access the cloud only run on the local host.

Reference:

[1] S. Sheng, M. Holbrook, P. Kumaraguru, L. F. Cranor, and J. Downs, "Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions," in Proceedings of the 28th international conference on Human factors in computing systems, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 373–382.

[2] B. Krebs, "HBGary Federal hacked by Anonymous," http: //krebsonsecurity.com/2011/02/hbgary-federal-hacked-by-anonymous/, 2011, accessed December 2011.

[3] B. Schneier, "Lockheed Martin hack linked to RSA's SecurID breach," http://www.schneier.com/blog/archives/2011/05/lockheed martin.html, 2011, accessed December 2011.

[4] C. Whittaker, B. Ryner, and M. Nazif, "Large-scale automatic classification of phishing pages," in NDSS '10, 2010.

[5] X. Dong, J. Clark, and J. Jacob, "Modelling user-phishing interaction," in Human System Interactions, 2008 Conference on, may 2008, pp. 627 –632.

[6] W. D. Yu, S. Nargundkar, and N. Tiruthani, "A phishing vulnerability analysis of web based systems," in Proceedings of the 13th IEEE Symposium on Computers and Communications (ISCC 2008). Marrakech, Morocco: IEEE, July 2008, pp. 326–331.

[7] Anti-Phishing Working Group (APWG), "Phishing activity trends report — second half 2010," http://apwg.org/reports/apwg report h2 2010. pdf, 2010, accessed December 2011.

[8] Anti-Phishing Working Group (APWG), "Phishing activity trends report — first half 2011," http://apwg.org/reports/apwg trends report h1 2011.pdf, 2011, accessed December 2011

PROBLEM STATEMENT DEFINISION:

The goal of our project is to implement a machine learning solution to the problem of detecting phishing and malicious web links. The end result of our project will be a software product which uses machine learning algorithm to detect malicious URLs. Phishing is the technique of extracting user credentials and sensitive data from users by masquerading as a genuine website. In phishing, the user is provided with a mirror website which is identical to the legitimate one but with malicious code to extract and send user credentials to phishers.

Phishing attacks can lead to huge financial losses for customers of banking and financial services. The traditional approach to phishing detection has been to either to use a blacklist of known phishing links or heuristically evaluate the attributes in a suspected phishing page to detect the presence of malicious codes.

The heuristic function relies on trial and error to define the threshold which is used to classify malicious links from benign ones. The drawback to this approach is poor accuracy and low adaptability to new phishing links. We plan to use machine learning to overcome these drawbacks by implementing some classification algorithms and comparing the performance of these algorithms on our dataset. We will test algorithms such as Logistic Regression, SVM, Decision Trees and Neural Networks on a dataset of phishing links from UCI Machine Learning repository and pick the best model to develop a browser plug in, which can be published as a chrome extension

3. IDEATION AND PROPOSED SOLUTION:

3.1. Empathy map canvas:

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.

FIG:EMPATHY MAP

3.2 Ideation & Brainstorming

Brainstorming is a group problem-solving method that involves the spontaneous contribution of creative ideas and solutions. This technique requires intensive, freewheeling discussion in which every member of the group is encouraged to think aloud and suggest as many ideas as possible based on their diverse knowledge.

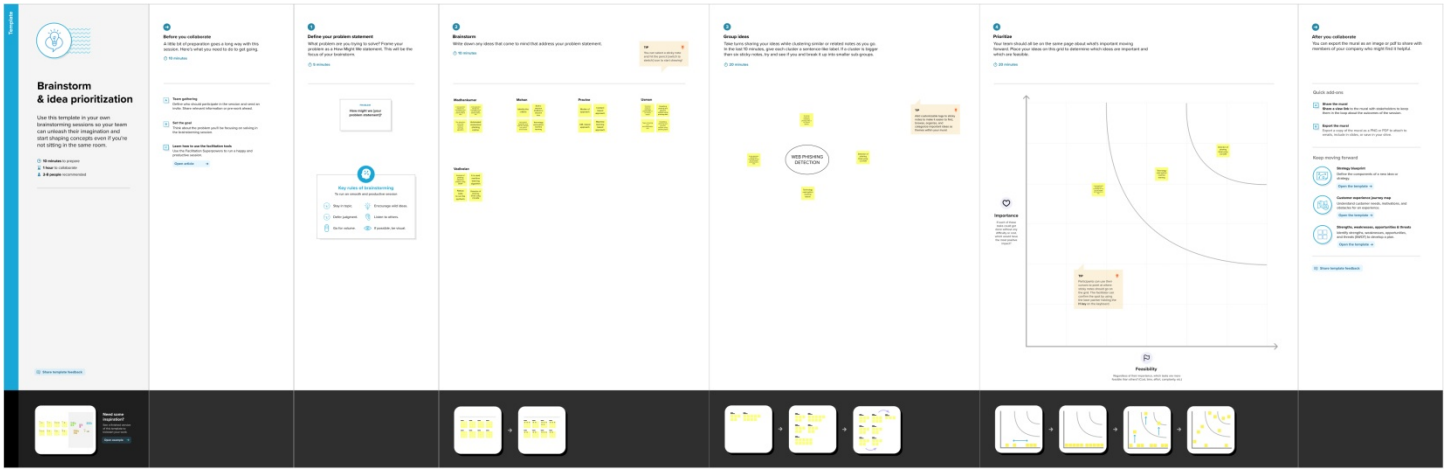


FIG:Brainstorm

3.3 PROPOSED SOLUTION

Problem Statement:

- ✓ Phishing is a fraudulent technique that is used over the internet to manipulate user to extract their personal information such as Username, Passwords, Credit Cards, Bank Account information etc.
- ✓ Phishing use multiple methods, including E-mail, Uniform Resource Locators(URL's), Instant messages, Form posting, Telephone calls and Text messages to steal user information.
- ✓ Many cypher infiltrations are accomplished through phishing attacks where user are tricked into interacting with web pages that appear to be legitimate.
- ✓ This project aim tto develop these methods of defense utilizing various approaches to categorising Websites and narrow them down to the best Machine Learning algorithm by comparing the accuracy rate, false positive and false negative rate of each algorithm.

Idea / Solution Description :

- ✓ This project aim to develop these methods of defense utilizing various approaches to categorising Websites and narrow them down to the best Machine Learning algorithm by comparing the accuracy rate, false positive and false negative rate of each algorithm.
- ✓ To find unknown malicious urls copared to the blacklist approach. iii. And Use anti-phishing protection and anti-spam software to protect yourself.

Novelty / Uniqueness :

- ✓ Our model uses the power of Machine learning to detect phishing sites.
- ✓ Python serves as a powerful tool to execute the application with Low false positives, High accuracy.
- ✓ Uses the latest techniques that gives an efficient and great performance.
- ✓ It can easily differentiate the fake and safe URL's. If it's fake means, a warning message will be intimate to the users.

Feasibility Of Ideas:

- ✓ Using data visualization and machine learning algorithm, we safeguard the user's data by detecting malicious websites.
- ✓ This application is easy to be built we have a lot of existing software tools that aid us in creating a web phishing detector.
- ✓ Faster, easier and seamless performance can be obtained. Business Model :
- ✓ Our model can be used by all user's to secure their data from malicious websites.

Social Impact :

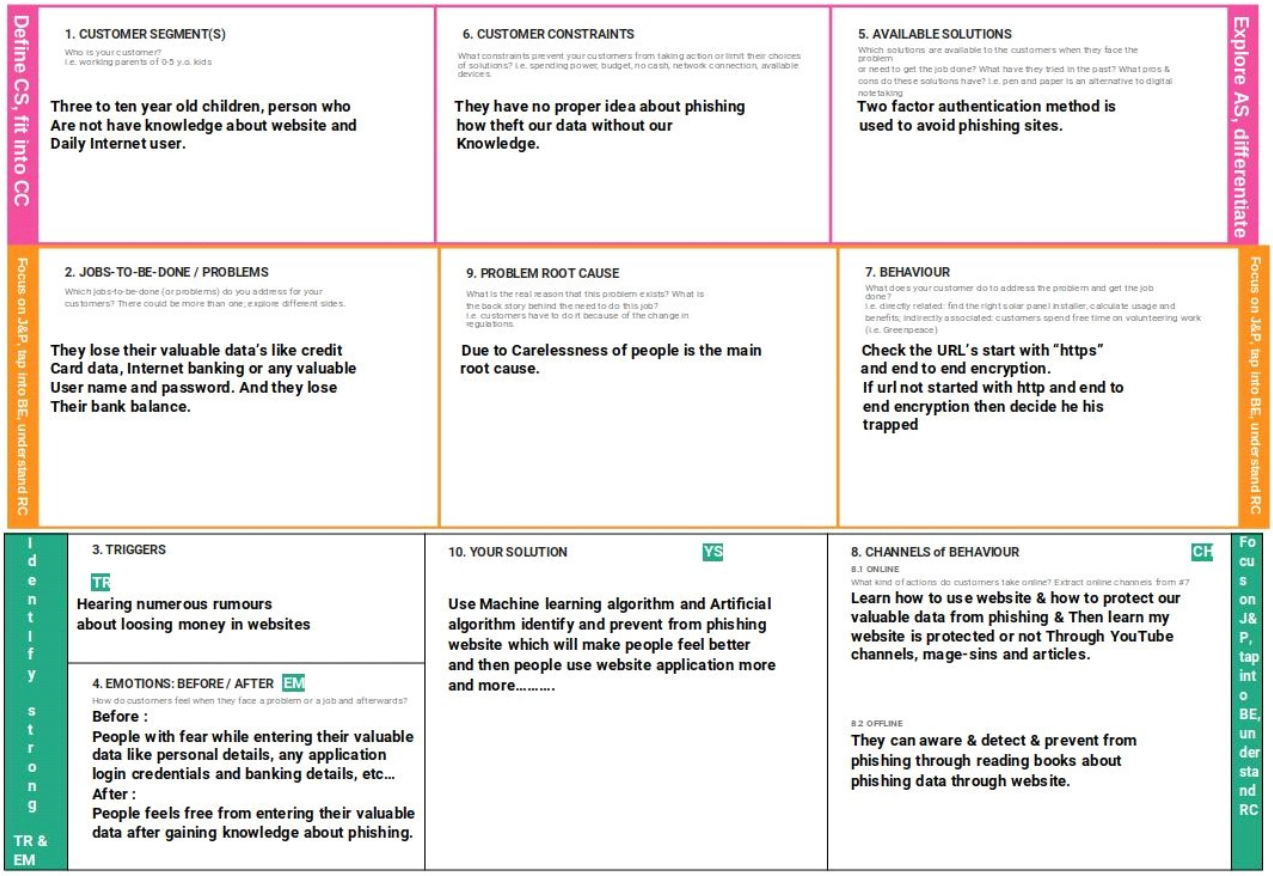
- ✓ According to recent research by Google, these was a 4505 increase in phishing websites from January to March 2021.
- ✓ Phishing has a list of negative effects on a business, including loss of money, loss of intellectual property, damage to reputation, and disruption of operational activities
- ✓ As an impact of this model, people can able to find fraudulent websites of fake ones.
- ✓ So that, they can avoid sharing sensitive data with unrecognized websites.

Scalability Of The Solution :

- ✓ A-part from E-Banking sector the idea proposed can be developed into platfrom independent model.
- ✓ Adapts to all sort of web application and ease of preventing users from scam

3.4 Problem Solution fit:

This occurs when you have evidence that customers care about certain jobs, pains, and gains. At this stage you've proved the existence of a problem and have designed a value proposition that addresses your customers' jobs, pains and gains.



4.REQUIREMENT ANALYSIS

There are two types of requirement,such as

- Functional requirement
- Non-functional requirement

4.1Functional requirement:

Functional requirements are the desired operations of a program, or system as defined in software development and systems engineering. The systems in systems engineering can be either software electronic hardware or combination software-driven electronics.

Following are the functional requirements of the proposed solution.

FR No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form.
FR-2	User Confirmation	Confirmation via Email.
FR-3	User Authentication	Authentication via Password.
FR-4	User Input	User input an URL to chech it is legal or phishing site
FR-5	Website Comparison	Model comparing the entered URL with the help of Blacklist and Whitelist
FR-6	Feature extraction	After comparing, if none found on comparison the it extracts feature using heuristic and visual similarity approach
FR-7	Prediction	Model Predicts the URL using Machine Learning algorithm such as Logistic Regression, KNN.
FR-8	Classifier	Model sends output to classifier and it produce final result
FR-9	Announcement	Model the displays whether the website is a legal or phishing site.
FR-10	Events	Model needs the capability of reetrieving and displaying accurate result for a website

Table:Functional Requirement

Non-functional requirement:

A non-functional requirement defines the quality attribute of a software system. It specifies “What should the software system do?” It places constraints on “How should the software system fulfill the functional requirements?”

FR No	Non-Functional Requirement	Description
NFR-1	Usability	A set of specifications that describe the system’s operation capabilities and constraints and attempt to improve its functionality.
NFR-2	Security	Assuring all data inside the system or its part will be protected against malware attacks or unauthorized access
NFR-3	Reliability	This approach gives more accuracy then existing system
NFR-4	Performance	Parameters for the proposed system gives accurate predicted value which is compared to the existing system.
NFR-5	Availability	The system is accessible by user at any time using web browser.
NFR-6	Scalability	The design will be suitable and performs with full efficieny according to rising demands

Table:Nonfunctional Requirement

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM:

It's easy to understand the flow of data through systems with the right data flow diagram software. This guide provides everything you need to know about data flow diagrams, including definitions, history, and symbols and notations. You'll learn the different levels of a DFD, the difference between a logical and a physical DFD and tips for making a DFD.

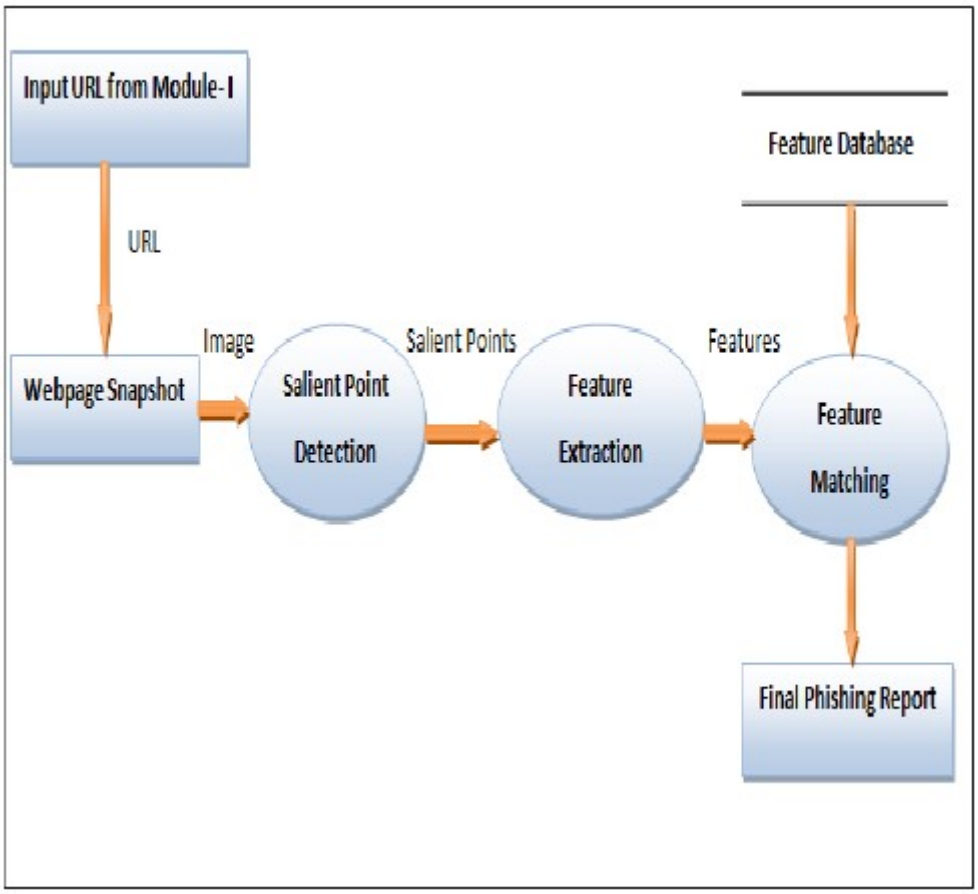


Fig Data flow Diagram

5.2 SOLUTION AND TECHNICAL ARCHITECTURE:

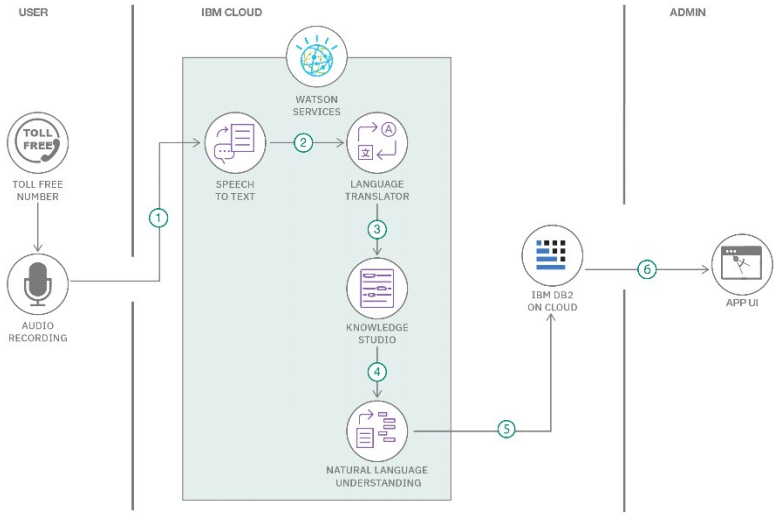
Solution architects oversee these tasks and activities and monitor a team's progress to keep the project on schedule. In contrast, technical architects complete the tasks to implement IT strategies. They ensure the solutions identified by other architects function correctly with the company's existing infrastructure.

TECHNOLOGY ARCHITECTURE:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/



5.3User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Dashboard	USN-1	As a user, I can easily navigate through dashboard and I can use the dashboard to get details about app and instruction to use the app.	Using dashboard i can easily access the application.	High	Sprint-1
	Url prediction and Result page	CCE-2	As a user, i can able to enter the URL to predict and View the corresponding result to that entered URL.	I can enter the URL and able to view the result	High	Sprint-2
	Add URL and Experience page and About page	USN-3	As a user, i can share my perviously experienced Phishing site and View about page of the website	I can add or enter experience and submit it	High	Sprint-3
Model Buliding	Prediction of Phishing sites	M-1	As an User, I can enter the url and Predict it as a Phishing site or not.	I can predict the URL is bad or good	High	Sprint-4
Model Testing	Testing of Model is worked as properly	MT-1	If the model Predict the URL as Phishing site or not with accuracy rate above 95%.		High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Member
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Madhankumar
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Mohan
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Usman ali
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Vadivelan
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Praveen
Sprint-2	Dashboard	USN-6	As a user, I can easily navigate through dashboard and I can use the dashboard to get details about app and instruction to use the app		Medium	Usman ali
Sprint-2	Customer Care Executive (Login)	CCE1	As a CCE I can login to application using User id & Password and I can interact with user	2	Medium	Mohan
Sprint-2	Customer Care Executive (Dasshboard)	CCE2	As a CCE I can access dashboard using User id and Password. I can see all user queries, explain app usage and rectify user	1	Low	Praveen
Sprint-3	Administrator (Login and Dashboard)	A-1	As an administrator, I can login and access dashboard and manage and direct activities.	1	High	Vadivelan
Sprint-3	Model Building	M-1	As an User, I can enter the url and Predict it as a Phishing site or not	2	High	Madhankumar
Sprint-4	Model Testing	MT-1	If the model Predict the URL as Phishing site or not with accuracy rate above 95%	3	High	Madhankumar

Table: Sprint Planning & Estimation

6.2 Sprint Delivery Schedule:

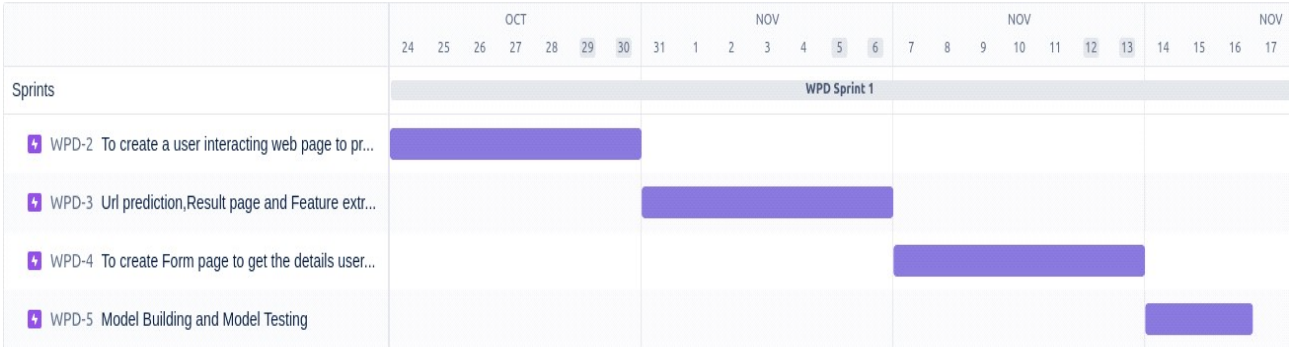
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.3 Reports from JIRA:



7. CODING & SOLUTIONING (Explain the features added in the project along with code):

1. Feature 1

The following stage is feature extraction, and that's an attribute extension that allows us to create more columns from URLs. Finally, we use a classifier algorithm to train our models. They take advantage of the obtained classified dataset. The remainder of our classified data would be used to validate the models. ML algorithms have been used to identify pre-processed data. That classifier utilized had been Random Forest

HTML AND JAVASCRIPT BASED FEATURE EXTRACTION:

1. Redirect

This number of cases a webpage is being rerouted seems to be the distinguishing factor between phishing and legitimate websites.

2. Right-Click disablement

JavaScript is used by phishers to block the right-click on a feature, preventing customers from accessing and purchasing website programs is written. This function is used at the same time that Using on Mouseover to Cover the Link is handled.

3. Making use of Pop-Up Window

It is really rare as for come across a malicious website that asks visitors to provide private details through a pop-up window.

4. Redirection of the I Frame

An I Frame is a type of HTML tag that allows you as for embed another website inside the one you're now viewing

Feature 2

That data contains several factors that should be considered when deciding whether a website URL is licit or phishing.

Address Bar based Features :

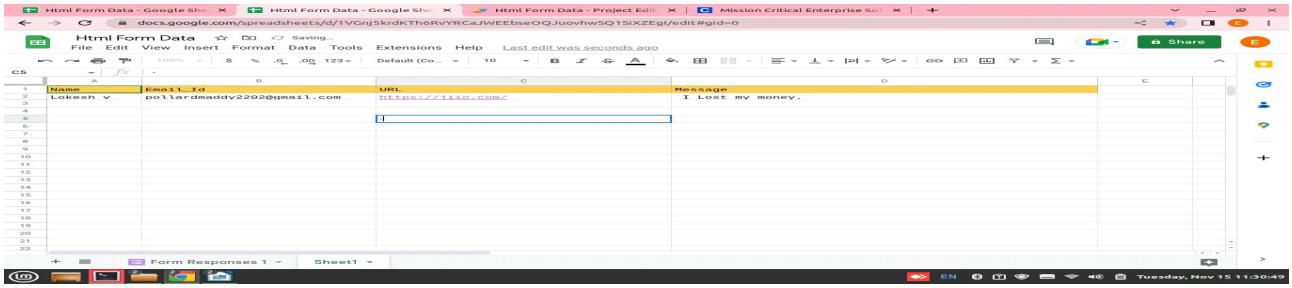
Using the IP address

If the URL has an IP address rather than a sphere name, such as 125.96.2.121, a person can practically be assured that his private detail are being stolen.

The Suspicious Part is hidden by a long URL By selecting a long URL, phishers can hide the suspect portion of the URL inside the URL bar.

Applying URL shortening services The URL is very short URL shortening is a mechanism on the Internet that allows a URL to be drastically reduced in length while still directing to the desired webpage.

2. Database Schema (if Applicable)



8. TESTING:

8.1 Test Cases:

Test Case ID	Test Case Description	Test Steps
TC01	Check Predict button is rooted to Prediction page	In home page, Click Prediction URL button.
TC02	In Prediction Page, Check prediction of url is done or not.	In prediction page, 1. Enter Url 2. Then press Prediction Button to predict URL
TC03	In Prediction output page, check the "Predict another URL" button.	In result page, press Predict another URL button.
TC04	In, Prediction Page, Check Prediction is done in positive and negative.	In prediction page, 1.Enter URL for good site and bad site. 2.then press Predict button.
TC05	Check User experience form is submitted in google form or not.	In add URL page, 1.Enter the Rrequired fields. 2.press submit button.
TC06	Check About button root to About page.	Press about button.
TC07	Check project Details button root's to Project details button.	Press Project details button
TC08	Check all buttons are working properly or not	Press all button and check it root's to corresponding page or not.

8.2 User Acceptance Testing:

8.2.1 Purpose of Document

This document is to briefly explain the test coverage and open issues of the Web Phishing Detection project at the time of the release to User Acceptance Testing (UAT).

8.2.2 Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
ByDesign	8	4	2	3	17
Duplicate	1	0	3	0	4
External	0	3	0	1	4
Fixed	9	2	4	15	30
NotReproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won'tFix	0	5	2	1	8
Totals	18	14	13	20	65

8.2.3 Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	TotalCases	NotTested	Fail	Pass
PrintEngine	2	0	0	2
ClientApplication	2	0	0	4
Security	1	0	0	1
OutsourceShipping	1	0	0	1
ExceptionReporting	1	0	0	1
FinalReportOutput	1	0	0	1
VersionControl	1	0	0	1

9.RESULT

9.1 Performance Metrics:

Our execution confirms that we had successfully implemented our project work and we had also tested them is different cases in the given timeline. Our project is distributes the work of design, implementation, testing and documentation in different levels so that we can complete our project on time. As the result, Our project Machine learning model predict Url is good or bad with 96% accuracy

10.ADVANTAGES :

- Measure the degrees of corporate and employee vulnerability.
- Eliminate the cyber threat risk level.
- Increase user alertness to phishing risks.
- Instill a cyber security culture and create cyber security heroes.

Disadvantage :

Phishing has a list of negative effects on a business, including loss of money, loss of intellectual property, damage to reputation, and disruption of operational activities. These effects work together to cause loss of company value, sometimes with irreparable repercussions.

11. CONCLUSION:

Our execution confirms that we had successfully implemented our project work and we had also tested them in different cases in the given timeline. Our project is distributes the work of design, implementation, testing and documentation in different levels so that we can complete our project on time. The results generated are up to the expected marks from which we concluded that our project is accomplised effectively, As a proof of completion we had produce the Demo video link and Coding of the project in our Documentation.

12. FUTURE SCOPE

We were planning to create a Google extension to predict whether a URL is good or bad

13.APPENDIX

Source Code

app_ibm.py

```
from flask import Flask,render_template,url_for,request

import inputScript

#import pymongo
from passlib.hash import  pbkdf2_sha256
import json
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "fVPrwgFk7x5q201PFtg4kKrYzBNHq5Ek7Nwiys5eCERk"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

app = Flask(__name__,template_folder='templates')


@app.route("/")
def helloworld():
    return render_template("/home.html")

@app.route("/predicturl")
def predicturl():
    return render_template("/predict1.html")


@app.route("/predict" ,methods=["POST","GET"] )

def predict():
    url = request.form['url']
    checkprediction = inputScript.main(url)

    print(url)
    print(checkprediction)


    # NOTE: manually define and pass the array(s) of values to be scored in the next line
    payload_scoring = {"input_data": [{"fields":
[['f0','f1','f2','f3','f4','f5','f6','f7','f8','f9','f9','f10','f11','f12','f13','f14','f15','f15','f16','f17','f18','f19','f20','f21','f22','f23','f24','f25','f26','f27']],
"values":checkprediction }]}

    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/62efb8db-e32e-4c70-bd7c-
7f819762d9b7/predictions?version=2022-11-12', json=payload_scoring,headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")
    print(response_scoring.json())
    pred = response_scoring.json()
    output = pred['predictions'][0]['values'][0][0]


    if output==1 :
```

```
        return render_template("/output1.html")

    elif output==-1 :
        return render_template("/output.html")

@app.route("/project_details")
def support():
    return render_template("/project_details.html")

@app.route("/addurl")
def addurl():
    return render_template("/addurl.html")

@app.route("/about")
def about():
    return render_template("/about.html")


if __name__ =="__main__":
    app.run(debug=True)
```

GITUP LINK:

https://github.com/IBM-EPBL/IBM-Project-41524-1660642606

DEMO VIDEO LINK:

https://youtu.be/JiNpTBxwd44