ANJUMAN-I-ISLAM'S KALSEKAR TECHNICAL CAMPUS
School of Engineering & Technology
Affiliated to : University of Mumbai, Recognised by : DTE (Maharashtra) & Approved by : AICTE (New Delhi)

1.1 Software Requirements Specification

for

Phishing Detection Chrome Extension

Version 1.0

Prepared by

Group Name: nullbits

Tauseef Shaikh	**18CO63**	**18CO63@aiktc.ac.in**
Sameer Shaikh	**19DCO06**	**19DCO06@aiktc.ac.in**
Sohail Sayyed	**18CO48**	**18CO48@aiktc.ac.in**

Instructor: **Prof. Kalpana Bodke Mam**

Course: **Software engineering**

Lab: ***CSL502***

Date: **04/03/2021**

CONTENTS I

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Initial and Version 1.0	Tauseef Shaikh/Sameer Shaikh	Initial Draft Version 1.0	04/03/21

1 Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

1.1 Document Purpose

The purpose of the document is for you to convey information about “Phishing Detection” and more importantly, for you to reflect on your experience and growth as a Web Developer or an aware User.
This document covers requirements, problem statements, technologies, and solutions to cybercrime phishing. It also explores the development of a chrome extension.

1.2 Product Scope

The Internet has widely spread all over the world covering every field of work. As a result, users who depend on the internet to carry out their businesses are also increasing considerably. This number tempts the imposters to carry out their fake operations. Eventually, end-users become more vulnerable to various kinds of web-attacks. One of the major implications of these web attacks affects the financial transactions over the internet. Phishing is one amongst the popular techniques that is used to gain the advantage of such security flaws. It is a cyberattack that is described as the art of mimicking a legitimate website of an authentic business targeting to gain access over its secretive information. These websites have extremely high graphical similarities to the real ones.

The scope for this research are
i. The investigation using the extension to detect phishing websites.
ii. The extension will focus on google chrome type of extension.
iii. The extension will use phishiTank as database

1.3 Intended Audience and Document Overview

Intended Audience of this document can be any security enthusiast, web developers, cyber security learners, students.

1.4 Definitions, Acronyms and Abbreviations

Phishing :- the fraudulent practice of sending emails or urls purporting to be from reputable companies in order to induce individuals to reveal personal information, such as passwords and credit card numbers

Chrome Extension :- Google Chrome extensions are programs that can be installed into Chrome in order to change or extend the browser's functionality. This includes adding new features to Chrome or modifying the existing behavior of the program itself to make it more convenient for the user.

URL	-	Uniform Resource Locator
DNS	-	Domain Name System
IP	-	Internet Protocol
MAC	-	Media Access Control
DOM	-	Document Object Model
API	-	Application Programming Interface
HTTP	-	HyperText Transfer Protocol

HTML	-	Hyper Text Markup Language
ARP	-	Address resolution Protocol

1.5 Document Conventions

Software Requirements Specification (SRS) based on ISO/IEC/IEEE 29148:2018 standard.

1.6 References and Acknowledgments

1 : - International Journal of Creative research Thoughts “ Detection of Phishing website Using Data Mining ”

Link :- Journal

2 : - Universiti malaysia Pahang “ WEBSITE DETECTION FOR PHISHING ATTACK BY USING BROWSER EXTENSION ”

Link :- Thesis template

2 Overall Description

2.1 Product Overview

Machine learning is a multidisciplinary approach initially used in supervised learning to form analytical models. It plays a major aspect in a broad scope of serious applications such as image recognition, data mining, skilled systems and image recognition. This approach appears suitable to solve phishing page detection, because this problem can be converted into a task of classification. ML techniques can be used to develop models to detect phishing activities based on categorizing old web pages and then these models can be integrated into the browser. Consider an example of a user browsing a web page, ML models will find the legitimate website instantly and then forward the output to the user at the other end. The vital factor for the success is the website's features in the input dataset and the availability of adequate websites for the creation of trustworthy analytical models, in developing ML models for automated anti-phishing identification.

We already learnt that, Phishing is a cyber-attack in which a person is made to visit illegal websites and fooled to reveal their hypersensitive data like name of user, bank details, card details, passwords etc. As primary security really matters on the web, phishing has drawn consideration of many experts and researchers. When there are two similar web pages, and information accompanied to the first page on apprehensive is entered by the user, an alert message should be raised on the second page second. When two web pages are not same, it is absurd that legitimate site is spoofed by second page, and thus the information can therefore be passed on without an alert that the page obtained is a legitimate page, based on keywords, by search done using a search engine or choosing between a set of predefined registered pages.

There are tools, capital of literature and methods for serving web users to recognise and refrain from phishing web pages. Some of the present phishing identification techniques are skilled in detecting phishing webpages with an extreme accuracy (>99%) while attaining extremely low accuracy of false classifying legitimate webpages (<0.1%). Although, a large number of these techniques, which make use of machine learning mainly depends on lots of inert characteristics, chiefly using the bag-of-words approach. As phishing identification methods struggle with gaining and upholding labelled data of training dataset. In accordance with deplorability perception, solutions which accordingly need minimum data for training are thereby very appealing [1,5,6]. Because of unavoidable phishing web pages mainly aiming at banks, online trading, governments and users of the web, it is necessary to avoid phishing attacks of web pages at the initial phase. Although, identification of a phishing web page is a laborious task, by virtue of the number of advanced approaches used by attackers to step out users of the web. The triumph of phishing web page identification techniques chiefly rely on identifying phishing web pages precisely and within an adequate period of time. As substitute solutions to the predictable phishing web page identification methods, a few inventive phishing identification methods are established and proposed in order to efficiently foresee phishing web pages. Over the last few years, the exceptional phishing web pages detection methods based on controlled machine learning techniques have been more often, which are more adaptive and clever to the atmosphere of the web associated with the predictable phishing web page identification methods.

The motivation in taking up the work is due to increasing phishing attacks from day to day and during the covid-19 pandemic it has doubled in numbers. According to the McAfee Covid-19 Threat Report, cyber criminals have been exploiting the pandemic through coronavirus-related malicious apps, phishing campaigns and malware, focusing on topics such as testing, treatments, cures and remote work. KnowBe4 reveals 56% of simulated phishing tests were related to coronavirus. Social media messages are another area of concern when it comes to phishing. Within the same report, KnowBe4's top-clicked social media email subjects reveal password resets, tagging of photos and new messages. Another example is the online classes taken on

various video call platforms where there is a high chance of someone posting an unknown link which might lead to phishing.

In this paper we make use of Random forest algorithm which is a collective learning technique for regression, classification and other tasks that works by creating an assembly of decision trees in a training set and ensuing in a class that is a mean prediction of the individual trees or the mode of the classes. The universal technique of random decision forests was first proposed by Tin Kam Ho in 1995. He emphasizes that forests of trees piercing with sloping hyperplanes as they can gain accuracy as they grow without being affected from overtraining, as long as the randomly limited forests are to be sensitive to only selected dimensions. The observation of a more complicated classifier obtained a more precision of monotonically sharp distinction to a collective belief that the complication of a classifier can solely raise to a point of accuracy before being offended by over fitting.

Background and existing system

They have developed a system that measures the conduct of the social architects, and a complete model for depicting mindfulness, estimation and resistance of social building-based assaults. They have proposed a hybrid multi-layered model utilizing normal language handling strategies for guarding the social designing-based assaults. The show empowers the fast recognition of a potential assailant attempting to control the unfortunate casualty for uncovering secrets. In this model they make use of a model named Security Training and Processing Evaluation (STPE) and this model contains a cycle with five stages. This model helps to protect the sensitive information from social engineering attacks.

In another method, they make use of a phishing location and anticipation method by joining URL-based and web page by similitude-based discovery. URL-based recognition includes selection of genuine URL (to which the site is actually coordinated) and the visual URL (which is identified by the client). This paper detects the phishing sites in two phases. The first phase is URL and Domain Identity Verification, in this phase we make use of LinkGuard algorithm to inspect the two URLs and then based on the result the procedure will proceed to the next stage. The second phase is image-based page matching. In this phase a snapshot of the original webpage and the suspicious web page will be taken, this is done either by the code developed or by utilizing a browser plug-in for webpage snapshot. Then they compare the snapshots, first they modify the image so that we have only less comparisons. They applied various transform methods like DFT (Discrete Fourier Transform), DCT (Discrete Cosine Transform) and other techniques like cross-correlation. If the detection of phishing sites are not detected by URL-based detection, then we make use of visual similarity-based detection. One of the novel techniques to check the site is legitimate or not.

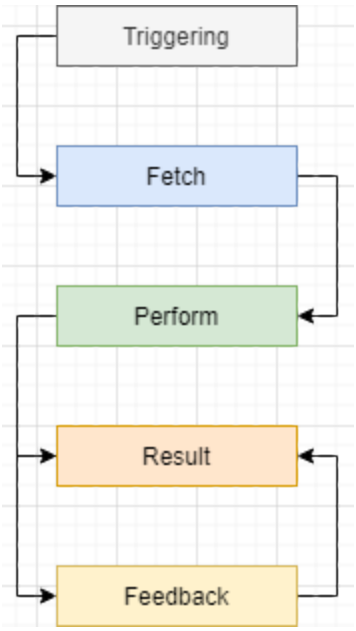
In other research work, the proposed system used secure QR code as an Anti-Phishing mechanism to stop web phishing. The system depends on the image captcha acceptance plan utilizing visual cryptography. It expects key and secret data from the phishing sites.

Waleed Ali proposed a procedure for detecting phishing websites by making use of supervised machine learning techniques such as radial basis function network (RBFN), naïve Bayes classifier (NB), back-propagation neural network (BPNN), decision tree, k-Nearest neighbour (kNN), random forest (RF) and support vector machine (SVM) a technique of detecting phishing website with wrapper features selection based on machine learning classifiers. In the research conclusions, the Neural Networks model was used in the process of classification, but it was prone to under fitting because it was poorly structured. However, it would overfit the training data set if structured to each single item in the dataset.

In this experimentation which is based on a number of features of the dataset which reveals that the self-structuring NN model was able to generate highly predictive anti-phishing models compared to other traditional C4.5 and probabilistic classification approaches.

The features which were considered include images, text pieces and styles, signature extraction, URL keywords and the overall appearance of the page as rendered by the browser were identified and considered for the experiment

2.2 Product Functionality



➔ **Triggering**

- Extension will be triggered automatically when user types any url or click on a url using chrome browser and also can be manually triggered by simply clicking on it

➔ **Fetch**

- then it will fetch the entered or pasted url content for further operations on it

➔ **Perform**

- firstly it will do normal domain lookup and then content lookup and then tags filter using ML

➔ **Result**

- Extension then will prepare result based on operations and will convert it into user friendly format

➔ **Feedback**

- Extension will also take feedback so it can then use those stats to improve the performance

2.3 Design and Implementation Constraints

Extensions can only do what is specified in chrome extension builder by abiding permission given by chrome browser that constrains the proposed system to track or detect phishing websites fully and with accuracy.

Phishing attacks can also change system files that cant be accessed through the extension proposed in this project.

2.4 Assumptions and Dependencies

Updated Chrome

Allowing access to chrome extensions

Not using conflicting things

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

This subsection of the SRS defines the UI requirements for phishing Detection Chrome extension System. The user interface for the extension shall be compatible with Chrome Browsers irrespective of OS.

3.1.2 Hardware Interfaces

All server-side components must execute on server-class computers. All client-side components must execute on workstation-class and personal-class computers.

3.1.3 Software Interfaces

The chrome extension should send the request with the user input data to the trained model on the server and return a prediction response based on the given input. The response should be made visible to the user after processing.

3.2 Functional Requirements

This section includes the requirements that specify all the fundamental actions of the software system.

3.2.1 User Class 1 - The User:

3.2.1.1 Functional Requirement 1.1:

ID: FR1
TITLE: User input URL.
DESC: The user should be able to fill a url to perform detection
These values when submitted will be sent to processing and predicting the outcome.

3.2.1.2 Functional Requirement 1.2:

ID: FR2
TITLE: Display the outcome.
DESC: The user will be able to see the predicted outcome returned by the model on the webpage

3.2.2 User Class 2 - Administrator:

3.2.2.1 Functional Requirement 2.1:

ID: FR3
TITLE: Data Collection
DESC: Data should be collected in many ways. Some data will be collected from phisTanks working in different organizations, some amount of data collected through Twitter API, some amount of data from college alumni databases.

3.2.2.2 Functional Requirement 2.2:

ID: FR4
TITLE: Data Pre-processing
DESC: The data collected will then have to be pre-processed before passing to train the model as this raw data will have some non-correlational columns and missing data. After this process of cleaning, it is then passed for OneHot encoding.

3.2.2.3 Functional Requirement 2.3:

ID: FR5
TITLE: OneHot Encoding
DESC: The OneHot encoder will convert the pre-processed data to numeric or other ordinal data format so that it can be passed further to train the model. This process is required for correct representation of distinct elements of a variable.

3.2.2.4 Functional Requirement 2.4:

ID: FR6
TITLE: Selecting Machine Learning Algorithm
DESC: Machine Learning Algorithms like SVM, Decision tree and XG boost should be tested for adequate accuracy of results. The one with optimum accuracy will be selected to train the final dataset.

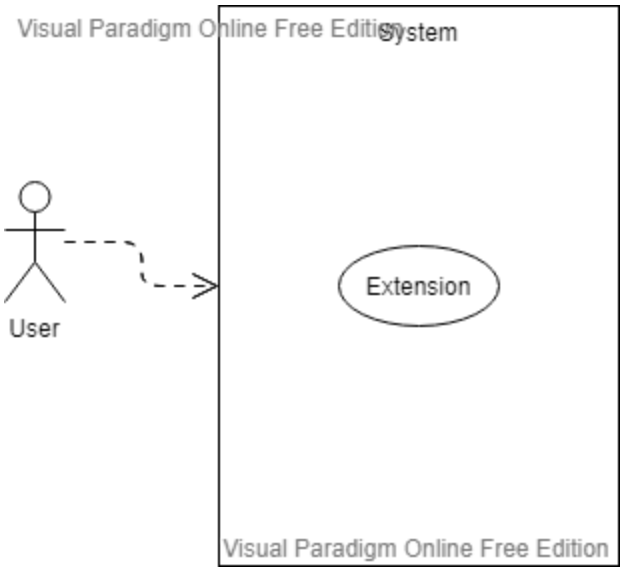
3.2.2.5 Functional Requirement 2.5:

ID: FR7
TITLE: Training and Testing
DESC: The model should be trained from the pre-processed data using suitable machine learning algorithms in order to make further predictions. Testing should be done from a small amount of already available dataset to check the performance of the model.

3.2.2.6 Functional Requirement 2.6:

ID: FR8
TITLE: Extension Integration
DESC: The trained model should be made available on a server where the extension will be hosted and should be connected to its backend. The user should be able to access this model by entering the input parameters and getting the prediction.

3.3 Use Case Model



4 Other Non-functional Requirements

4.1 Performance Requirements

The system must be interactive and the delays involved must be less. So, in every action-response of the system, there are no immediate delays. Also, when connecting to the server the delay is based on editing on the distance of the 2 systems and the configuration between them so there is high probability that there will be or not a successful connection in less than 20 seconds for sake of good communication. The application should load and be usable within 3 seconds. The application should update the interface on interaction within 5 seconds

4.2 Safety and Security Requirements

The user can use the features of the extension without login through his personal details so it provides anonymity and unbiased results. The entered data is only used for prediction purposes and will not be stored in the database.