HIGH LEVEL DESIGN (HLD)

PHISHING DOMAIN DETECTION (PDD)

Revision Number: 2.0

Last date of revision: 12-03-2023

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| 05/001/2023 | 1 | Initial HLD – V1.0 | SK mahammadrafi |
| 12/03/2023 | 2 | Updated model | SK mahammadrafi |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

[Document Version Control 2](#_heading=h.gjdgxs)

[Abstract 4](#_heading=h.30j0zll)

1. [Introduction 5](#_heading=h.1fob9te)
   1. [Why this High-Level Design Document? 5](#_heading=h.3znysh7)
   2. [Scope 5](#_heading=h.2et92p0)
   3. [Definitions 5](#_heading=h.tyjcwt)
2. [General Description 6](#_heading=h.3dy6vkm)
   1. [Product Perspective 6](#_heading=h.1t3h5sf)
   2. [Problem statement 6](#_heading=h.4d34og8)
   3. [PROPOSED SOLUTION 6](#_heading=h.2s8eyo1)
   4. [FURTHER IMPROVEMENTS 6](#_heading=h.17dp8vu)
   5. [Technical Requirements 6](#_heading=h.3rdcrjn)
   6. [Data Requirements 7](#_heading=h.26in1rg)
   7. [Tools used 8](#_heading=h.lnxbz9)
      1. [Hardware Requirements 8](#_heading=h.35nkun2)
      2. [ROS(Robotic Operating System) 9](#_heading=h.1ksv4uv)
   8. [Constraints 9](#_heading=h.44sinio)
   9. [Assumptions 9](#_heading=h.2jxsxqh)
3. [Design Details 10](#_heading=h.z337ya)
   1. [Process Flow 10](#_heading=h.3j2qqm3)
      1. [Model Training and Evaluation 10](#_heading=h.1y810tw)
      2. [Deployment Process 11](#_heading=h.4i7ojhp)
   2. [Event log 11](#_heading=h.2xcytpi)
   3. [Error Handling 11](#_heading=h.1ci93xb)
   4. [Performance 12](#_heading=h.3whwml4)
   5. [Reusability 12](#_heading=h.2bn6wsx)
   6. [Application Compatibility 12](#_heading=h.qsh70q)
   7. [Resource Utilization 12](#_heading=h.3as4poj)
   8. [Deployment 12](#_heading=h.1pxezwc)

# Abstract

Phishing is a type of fraud in which an attacker impersonates a reputable company or

person in order to get sensitive information such as login credentials or account

information via email or other communication channels. Phishing is popular among

attackers because it is easier to persuade someone to click a malicious link that appears

to be authentic than it is to break through a computer's protection measures..

# Introduction

## Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

* + - Present all of the design aspects and define them in detail
    - Describe the user interface being implemented
    - Describe the hardware and software interfaces
    - Describe the performance requirements
    - Include design features and the architecture of the project
    - List and describe the non-functional attributes like:
      * Security
      * Reliability
      * Maintainability
      * Portability
      * Reusability
      * Application compatibility
      * Resource utilization
      * Serviceability

## Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

## Definitions

|  |  |
| --- | --- |
| *Term* | *Description* |
| *PDD* | Phishing Domain Detection |
| *Database* | Collection of data from phishing website and dumping into mongo db data base , collect from mongo db database |
| *IDE* | Integrated Development Environment |
| *AWS* | Amazon Web Services |

# General Description

## Product Perspective

## The phishing domain website solution system is a supervised machine learning , classification model which will help us to detect is the website legitimate or not.

#### Problem statement

To create an supervised classification model using phishing domain website to implement the following use cases.

* + - Whether the website is legitimate or not.
    - Computer security enthusiasts can find these datasets interesting for building firewalls, intelligent ad blockers, and malware detection systems.

#### PROPOSED SOLUTION

The solution proposed here is an PDD (Phishing Domain Detection) based Surveillance (Phishing Domain Detection) can be implemented to perform above mention use cases .

Phishing involves attempts by Internet fraudsters to access and obtain personal and sensitive information, such as usernames, passwords, and financial information, by utilizing social engineering techniques. Typically a victim receives a message that appears to have been sent by a known contact or organization. The message contains malicious software targeting the user’s computer or has links to direct victims to malicious websites in order to trick them into divulging personal and financial information, such as passwords, account IDs or credit card details.

#### FURTHER IMPROVEMENTS

These days all works and everything is are digitalized in every domain, for hackers it is easy to hacking sensitive information and credentials form the people , to overcome these type of problem theses data will helpful.

## Data Requirements

There are a lot of algorithms and a wide variety of data types for phishing detection in the academic literature and commercial products. A phishing URL and the corresponding page have several features which can be differentiated from a malicious URL. For example; an attacker can register long and confusing domain to hide the actual domain name **(**Cybersquatting, Typosquatting). In some cases attackers can use direct IP addresses instead of using the domain name

These data consist of a collection of legitimate, as well as phishing website instances. Each website is represented by the set of features that denote whether the website is legitimate or not. Data can serve as input for the machine learning process.

 Features collected from academic studies for the phishing domain detection with machine learning techniques are grouped as given below.

1. URL-Based Features
2. Domain-Based Features
3. Page-Based Features
4. Content-Based Features

## URL-Based Features

URL is the first thing to analyse a website to decide whether it is a phishing or not. As we mentioned before, URLs of phishing domains have some distinctive points. Features which are related to these points are obtained when the URL is processed. Some of URL-Based Features are given below.

* Digit count in the URL
* Total length of URL
* Checking whether the URL is Typosquatted or not. (google.com → goggle.com)
* Checking whether it includes a legitimate brand name or not (apple-icloud-login.com)
* Number of subdomains in URL
* Is Top Level Domain (TLD) one of the commonly used one?

## Domain-Based Features

The purpose of Phishing Domain Detection is detecting phishing domain names. Therefore, passive queries related to the domain name, which we want to classify as phishing or not, provide useful information to us. Some useful Domain-Based Features are given below.

* Its domain name or its IP address in blacklists of well-known reputation services?
* How many days passed since the domain was registered?
* Is the registrant name hidden?

## Page-Based Features

Page-Based Features are using information about pages which are calculated reputation ranking services. Some of these features give information about how much reliable a web site is. Some of Page-Based Features are given below.

* Global Pagerank
* Country Pagerank
* Position at the Alexa Top 1 Million Site

Some Page-Based Features give us information about user activity on target site. Some of these features are given below. Obtaining these types of features is not easy. There are some paid services for obtaining these types of features.

* Estimated Number of Visits for the domain on a daily, weekly, or monthly basis
* Average Pageviews per visit
* Average Visit Duration
* Web traffic share per country
* Count of reference from Social Networks to the given domain
* Category of the domain
* Similar websites etc.

## Content-Based Features

Obtaining these types of features requires active scan to target domain. Page contents are processed for us to detect whether target domain is used for phishing or not. Some processed information about pages are given below.

* Page Titles
* Meta Tags
* Hidden Text
* Text in the Body
* Images etc.

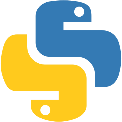
By analysing these information, we can gather information such as;

* Is it required to login to website
* Website category
* Information about audience profile etc.

All of features explained above are useful for phishing domain detection. In some cases, it may not be useful to use some of these, so there are some limitations for using these features. For example, it may not be logical to use some of the features such as Content-Based Features for the developing fast detection mechanism which is able to analyze the number of domains between 100.000 and 200.000. Another example would be, if we want to analyze new registered domains Page-Based Features is not very useful. Therefore, the features that will be used by the detection mechanism depends on the purpose of the detection mechanism. Which features to use in the detection mechanism should be selected carefully.

## Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, matplotlib are used to build the whole model.



* + - PyCharm is used as IDE.
    - For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
    - AWS is used for deployment of the model.
    - Tableau/Power BI is used for dashboard creation.
    - MySQL/MongoDB is used to retrieve, insert, delete, and update the database.
    - Front end development is done using HTML/CSS
    - Python Django is used for backend development.
    - GitHub is used as version control system.

## Constraints

The Phishing domain detection solution system must be user friendly, as automated as possible and users should not be required to know any of the workings.

## Assumptions

## Phishing is a form of fraud in which the attacker tries to learn sensitive information such as login credentials or account information by sending as a reputable entity or person in email or other communication channels.

# Design Details

## Process Flow

Below is the process flow diagram is as shown below.

### Model Training and Evaluation

New data

Train dataset

Dataset

New data prediction

Model building

EDA and PREPROCESSING

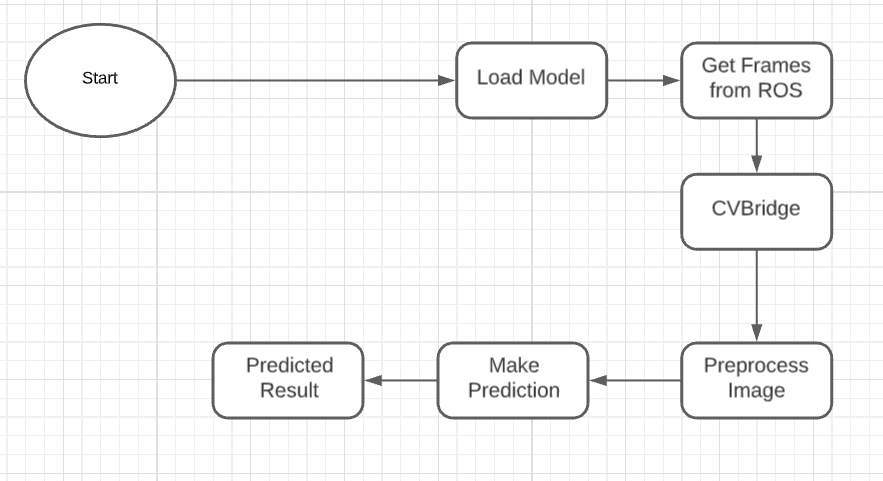
Test dataset

Training

solitting

Evaluation

### Deployment Process



## Event log

The system should log every event so that the user will know what process is running internally.

**Initial Step-By-Step Description:**

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

## Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

# Performance

# Phishing is a type of fraud in which an attacker impersonates a reputable company or

# person in order to get sensitive information such as login credentials or account

# information via email or other communication channels. Phishing is popular among

# attackers because it is easier to persuade someone to click a malicious link the appears to be authentic than it is to break through a computer's protection measures.

## Reusability

The code written and the components used should have the ability to be reused with no problems.

## Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

## Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

* 1. **Deployment**



# 