Low Level Design (LLD)

Phishing Domain Detection

Revision Number: 1
Last date of revision: 30/03/2023

P Santosh Singh

• Document Version Control

Date Issued	Version	Description	Author
30th Mar 2023	1.1	First Draft	Santosh Singh

Contents

Abstract

Phishing stands for a fraudulent process, where an attacker tries to obtain sensitive information from the victim. Usually, these kinds of attacks are done via emails, text messages, or websites. Phishing websites, which are nowadays in a considerable rise, have the same look as legitimate sites. However, their backend is designed to collect sensitive information that is inputted by the victim.

This work discusses the implementation of a Machine learning algorithm to identify potential phishing attempts using the URL shared by the users and help them understand if they'll be safe if the link is opened.

We won't be opening the link to identify a potential phishing URL.

1.Introduction

1.1 Why this Low-Level Design Document?

The purpose of this Low-Level Design (LLD) 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 main objective of the project is to identify harmful/malicious URL and safeguard user from being Phished. The Phishing domain detection project can help:

- Capture malicious URL.
- 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

a. 1.2 Scope

The LLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The LLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system. This software system will be a Web application This system will be designed to detect Phishing attempts using URLs.

b. 1.3 Constraints

We will only be detecting Phishing attempts using URLs.

c. 1.4 Risks

Document specific risks that have been identified or that should be considered.

d. 1.5 Out of Scope

Delineate specific activities, capabilities, and items that are out of scope for the project.

2. Technical specifications

2.1 Dataset

- dataset_full.csv
  Full variant - dataset_full.csv
  a. Short description of the full variant dataset:
  b. Total number of instances: 88,647
  c. Number of legitimate website instances (labeled as 0): 58,000
  d. Number of phishing website instances (labeled as 1): 30,647
  e. Total number of features: 111
- dataset_small.csv
  Small variant - dataset_small.csv
  a. Short description of the small variant dataset:
  b. Total number of instances: 58,645
  c. Number of legitimate website instances (labeled as 0): 27,998
  d. Number of phishing website instances (labeled as 1): 30,647
  e. Total number of features: 111

2.2 Logging

We should be able to log every activity done by the incidents.

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

a. 2.5 Database

System needs to store every request into the database and we need to store it in such a way that it is easy to retrain the model as well.

1. The User chooses the activity dataset.

2. The User gives required information.
3. The system stores each and every data given by the user or received on request to the database. Database you can choose your own choice whether MongoDB/ MySQL.

3. Deployment

4.Technology stack

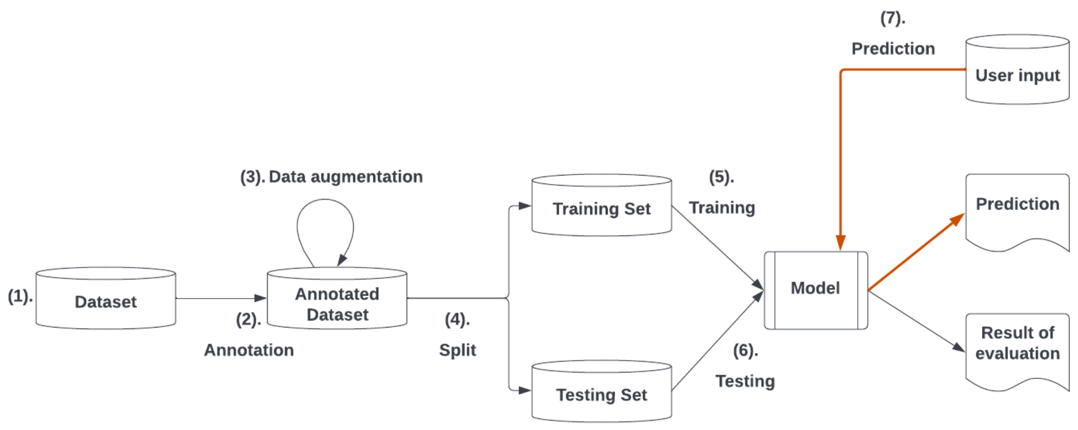
Front End	HTML/CSS/JSt
Backend	Python Django
Database	MongoDB/MySql
Deployment	AWS
Visualization	Matplotlib,Seaborn ,Plotly
Dashboard	NA
version control	GitHub

5. Proposed Solution

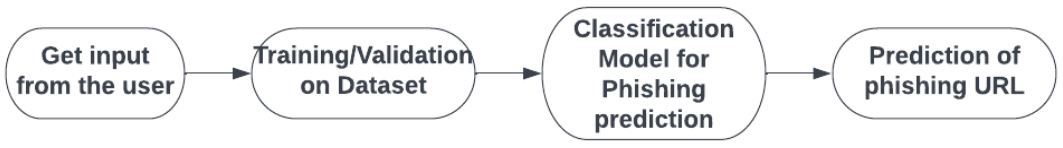
The designed phishing domain detection model will detect harmful/malicious URL based on various URL features, so that we can identify URL which should be interacted with and safeguard the user data.

1. Baseline Model: RandomForest Classifier.

6.Model training/validation workflow



7 .User I/O workflow


Get input from the user
Training/Validation on Dataset
Classification Model for Phishing prediction
Prediction of phishing URL

8. 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.

9.Test cases

Attempt	Model	Accuracy	Details
3rd attempt	RandomForest	85%	Hyper-parameter tuning
2nd attempt	RandomForest	70%	feature engineering applied
1st attempt	RandomForest	50%	no feature engineering applied

10. Conclusion

The designed phishing domain detection model will detect harmful/malicious URL based on various URL features, so that we can identify URL which should be interacted with and safeguard the user data.