High Level Design (HLD)

(Phishing Domain Detection)

Written By	Abhijeet R. Waghchaure
Document Revision Number	1.0
Last Date of Revision	15/08/2022

Document Version Control

Date Issued		Version	Description	Author
09/08/2022		1	Initial HLD – V1.0	Abhijeet

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.

Discovering and detecting phishing websites has recently also gained the machine learning community's attention, which has built the models and performed classifications of phishing websites. This paper presents two dataset variations that consist of 58,645 and 88,647 websites labeled as legitimate or phishing and allow the researchers to train their classification models, build phishing detection systems, and mining association rules.

1 Introduction

1.1 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

1.2 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.

1.3 Definitions

Term	*Description*
PDD	Phishing Domain Detection
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment
AWS	Amazon Web Services

2 General Description

2.1 Product Perspective

We have to build a solution that should able to predict whether the domain is real or fake.

2.2 Problem Statement

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.

The mail goal is to predict whether the domains are real or malicious.

2.3 Proposed Solution

The solution proposed here is a Phishing Domain detection can be implemented to perform above mention use cases. In first case, if where an attacker tries to obtain sensitive information from the victim. Usually, these kinds of attacks are done via emails, text messages, or websites. Discovering and detecting phishing websites has recently also gained the machine learning community’s attention, which has built the models and performed classifications of phishing websites.

2.4 Further Improvements

Solution can be added with more use cases like email spam detection.

2.5 Data Requirements

The presented dataset was collected and prepared for the purpose of building and evaluating various classification methods for the task of detecting phishing websites based on the uniform resource locator (URL) properties, URL resolving metrics, and external services. The attributes of the prepared dataset can be divided into six groups:

- Attributes based on the whole URL properties
- Attributes based on the domain properties
- Attributes based on the URL directory properties
- Attributes based on the URL file properties
- Attributes based on the URL parameter properties
- Attributes based on the URL resolving data and external metrics

2.6 Value of the Data:

- 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.

- Machine learning and data mining researchers can benefit from these datasets, while also computer security researchers and practitioners. Computer security enthusiasts can find these datasets interesting for building firewalls, intelligent ad blockers, and malware detection systems.

- This dataset can help researchers and practitioners easily build classification models in systems preventing phishing attacks since the presented datasets feature the attributes which can be easily extracted.

- Finally, the provided datasets could also be used as a performance benchmark for developing state-of-the-art machine learning methods for the task of phishing websites classification.

2.6 Tools used

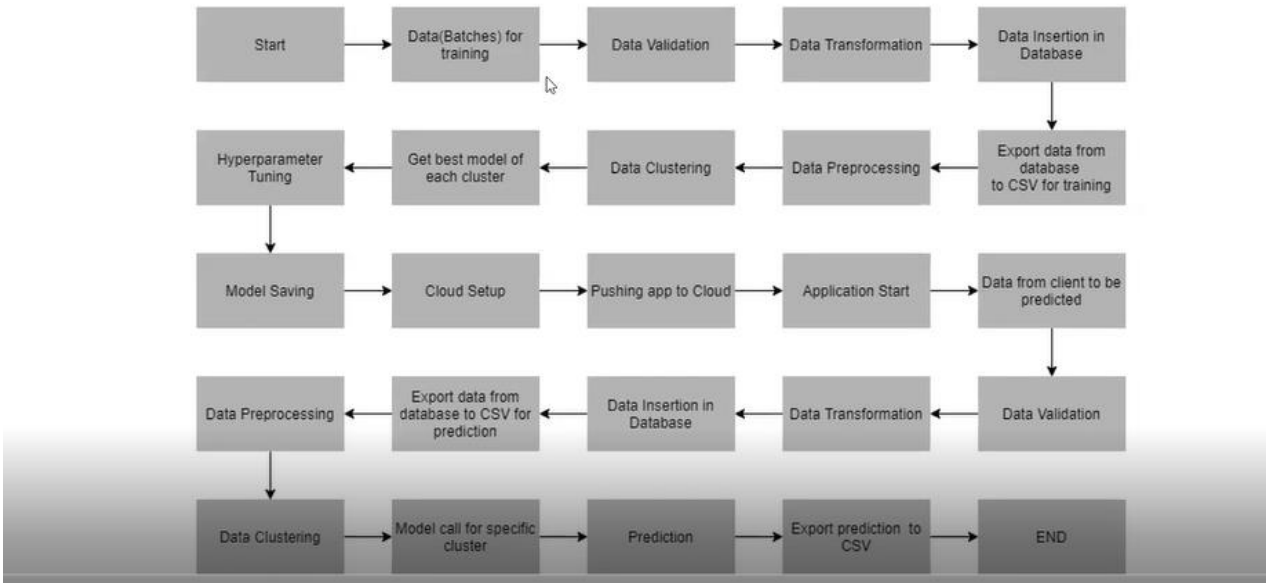
- PyCharm is used as IDE.
- For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
- AWS is used for deployment of the model.
- Tableau/Power BI is used for dashboard creation.
- MySQL/MongoDB is used to retrieve, insert, delete, and update the database.
- Front end development is done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

3 Design Details

3.1 Process Flow

For identifying the different types of anomalies, we will use a deep learning base model. Below is the process flow diagram is as shown below.

3.1.1 Application Flow:



3.2 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.

3.3 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.

3.4 Performance:

Two dataset variations that consist of 58,645 and 88,647 websites labeled as legitimate or phishing and al- low the researchers to train their classification models, build phishing detection systems, and mining association rules. Also, model retraining is very important to improve the performance.

3.4 Reusability

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

3.5 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.

3.6 Resource Utilization

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

3.7 Deployment

4. Conclusion

- The final take away from this project is to explore various machine learning models, perform Exploratory Data Analysis on phishing dataset and understanding their features.
- Creating ipynb notebook or .py files will help me to learn a lot about the features affecting the models to detect whether URL is safe or not, also I will came to know how to tuned model and how they will affect the model performance.
- The final conclusion on the Phishing dataset is that the some feature like "HTTTPS", "AnchorURL", "WebsiteTraffic" will have more importance to classify URL is phishing URL or not.
- Some Classifiers correctly classify URL up to some percentage with respective to classes and hence reduces the chance of malicious attachments.

5 References

1. https://www.sciencedirect.com/science/article/pii/S2352340920313202

2. https://drive.google.com/file/d/19xPfUcVq1shDiiri3V3PUBEE9LEBvEFM/view

3. https://towardsdatascience.com/phishing-domain-detection-with-ml-5be9c99

THE END