High Level Design (HLD)

Phishing Attack Domain Detection

Revision Number: 1.0

Last date of revision: 15/12/2021

Document Version Control

Date Issued	Version	Description	Author
15/12/2021	1	Initial HLD – V1.0	Deepesh mhatre

Contents

Abstract

We live in the digital era. This has steadily changed the way you buy things, pay your bills, rent an apartment, watch a movie, and everything else. All of this is made possible, because of the internet and electronic media.Due to this widespread use, there are innumerable incidents of a security breach, fraud,malicious attacks, etc reported. To keep the internet age well-ordered and safe for users, the need for Cybersecurity arises. It secures you from Cyber-criminals, fraudsters, hackers, and anybody who wants to harm you financially, mentally, or engage in data theft online.

This project focuses on solving a most common cyber fraud called phishing.Phishing is a type of social engineering attack often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message. The recipient is then tricked into clicking a malicious link, which can lead to the installation of malware, the freezing of the system as part of a ransomware attack or the revealing of sensitive information.In this project we discuss about creating a machine learning system which can detect malicious url links and how we can prevent phishing attacks using it.

.

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*
Phishing	An online cyber attack
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

The Phishing Attack Domain Detection system is a deep learning-based regression model which will help us to predict if Urls/links are malicious or designed for phishing attacks and warn users before clicking them.

2.2 Problem statement

To create an AI solution for detecting if a given URL or a link is malicious or designed for phishing attacks which retrieve user's data for unethical use. Following are the use-cases where this solution can be applied :

- Search engines & browsers can warn users about malicious links
- Social media sites can prevent users from posting/clicking malicious links
- This can also be used to find the credibility of a web domain

2.3 PROPOSED SOLUTION

The above mentioned problem statement can be solved by Implementing a deep learning based regression model. On a high level the working of the model is simple.The model will take a URL as input and predict the probability of the URL being malicious. The model will look at various other features & characteristics of the domain to determine if the URL is malicious or legitimate.

2.4 FURTHER IMPROVEMENTS

The phishing-domain-detection system can also be integrated with other large scale cyber-security systems. It can act as a microservice in a large scale application's architecture.

2.5 Technical Requirements

This document addresses the requirements for creating an AI system for detecting malicious URLs. The system needs to be optimal and should be fail safe. Some of the requirements to build such solution are the following.

- Data on Legitimate & Malicious URLs
- Appropriate Tools to analyze that data
- GPU optimized Computer hardware for training
- Cloud providers to deploy the system online

2.6 Data Requirements

Data requirements completely depend on our problem statement.

- We need URL data of both legitimate & malicious urls
- The dataset must be balanced i.e must contain equal amount of positive and negative classes.
- The data must contain diverse features and characteristics with less sparsity among features.
- The data must contain both numerical and categorical features in balance.
- The dataset must be verified and if possible contain real-life scenarios for optimal results

2.7 Tools used

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

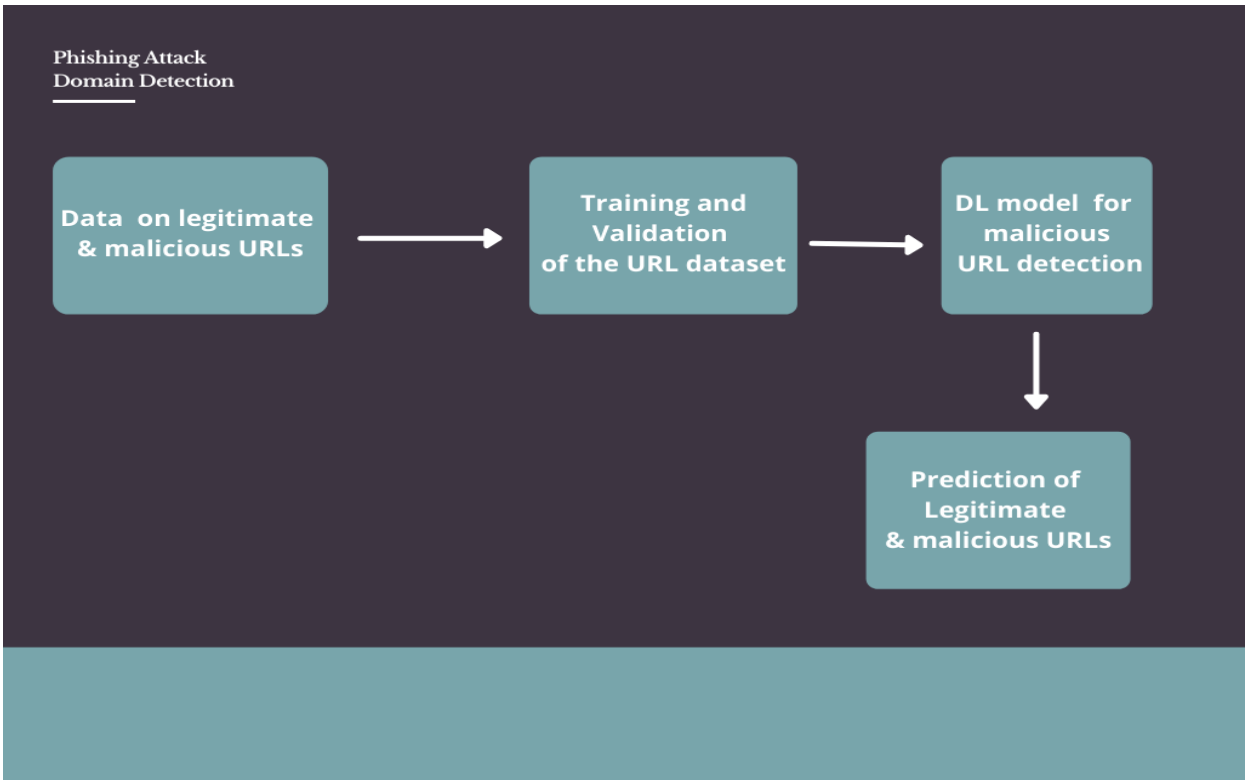
- PyCharm is used as IDE.
- For visualization of the plots, Matplotlib,Seaborn and Plotly are used.
- Heroku is used for deployment of the model.
- Front end development is done using React & Css
- Python FastAPI is used for backend development.
- GitHub is used as version control system.

3 Design Details

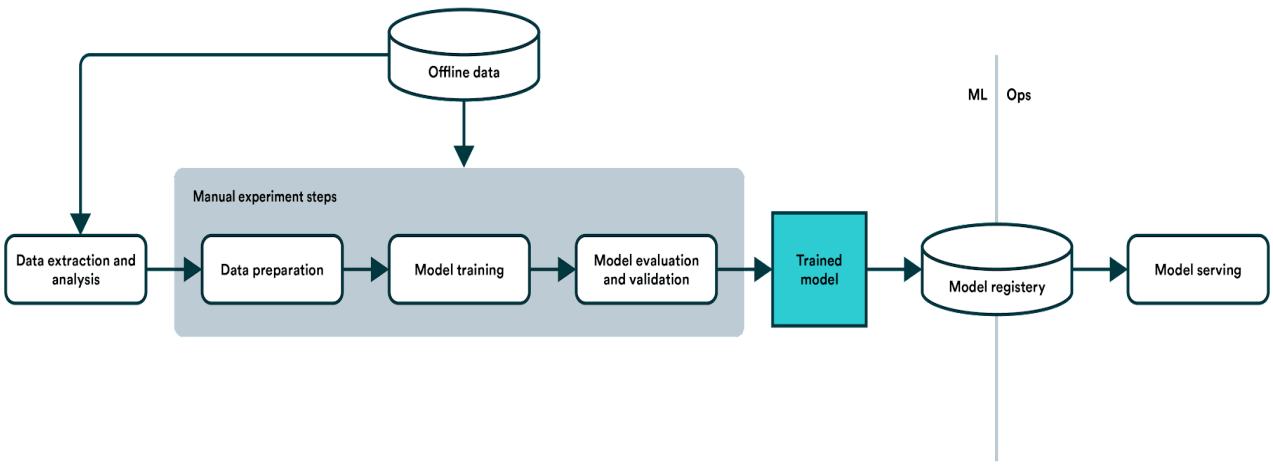
3.1 Process Flow

For identifying the malicious URLs, we will use a deep learning base model. Below is the process flow diagram is as shown below.

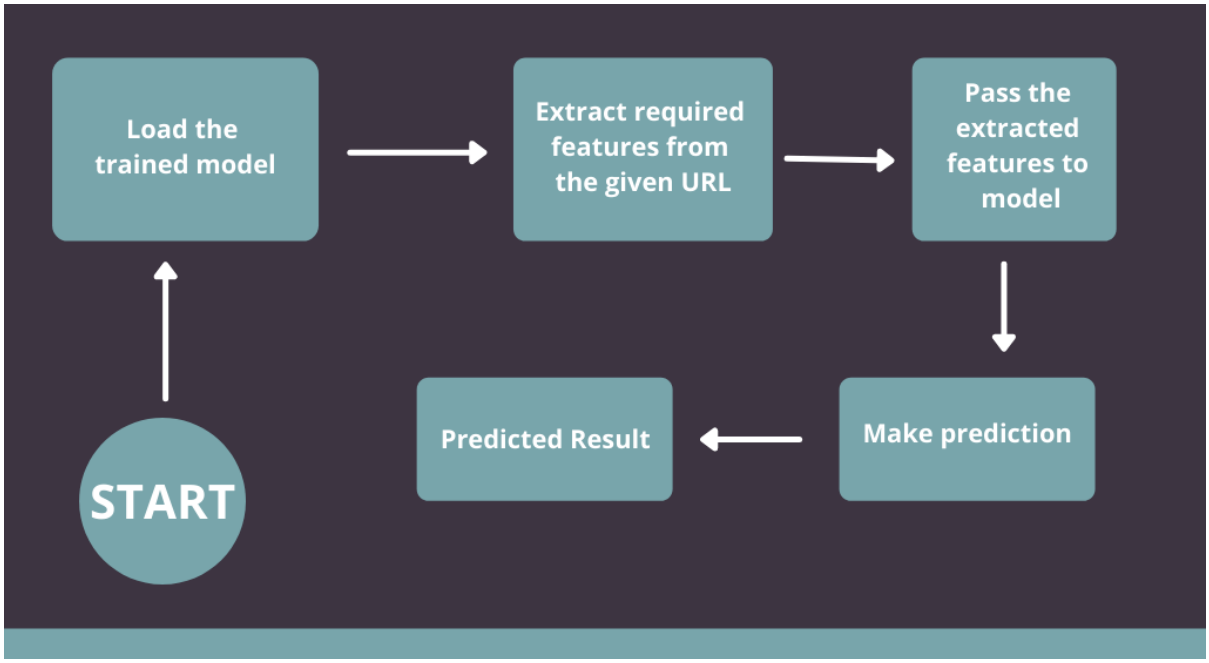
Proposed methodology



3.1.1 Model Training and Evaluation



3.1.2 Deployment Process



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.

4 Performance

4.1 Reusability

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

4.2 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.

4.3 Resource Utilization

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

4.4 Deployment

5 Conclusion

The Designed deep-learning based Phishing domain detection system takes a string URL as input and returns a probability value (0-100) of URL being malicious. We declare a URL malicious if it crosses a probability value of 70%. It is deployed as both REST API and a web interface has also been created to interact with it.

6 References

1. https://en.wikipedia.org/wiki/Phishing
2. Google.com for images.