Chrome Extension for Detecting Phishing Websites:

Abstract:

The "Phishing Site Detector Plugin" is a lightweight Chrome extension designed to identify and warn users about potential phishing websites without compromising their browsing privacy. Unlike traditional methods that rely on server-side analysis, this plugin operates entirely on the client side, ensuring user data remains confidential. The core functionality utilizes a pre-trained Random Forest Classifier model, downloaded once during installation, to classify websites in real-time.

This project is mainly of implementing the in Javascript for it to run as a browser plugin. Since javascript doesn’t have much ML libraries support and considering the processing power the client machines, the approach needs to be made lightweight.

**Introduction :** Phishing attacks pose a significant threat to online security, targeting users through deceptive websites that mimic legitimate ones. Detecting such fraudulent websites promptly is critical to prevent data theft and financial loss.

**Problem Statement and Overview :** Existing phishing detection solutions often involve server-side processing, potentially compromising user privacy by requiring access to browsing data. The challenge lies in developing a robust client-side solution that can effectively identify phishing websites without invasive data collection.

**Tools and Applications** **:**  The plugin leverages Python 3.7 ,& the scikit-learn library (specifically sklearn==0.19.2) for implementing a Random Forest Classifier. This classifier is trained using a dataset sourced from the UCI Repository, ensuring a diverse range of phishing patterns are captured.

|  |
| --- |
| Install scikit-learn version 0.19.2,  Install numpy version 1.15.0 ,  Install liac-arff version 2.2.2 |

**Dataset:** [UCI Repository](https://archive.ics.uci.edu/ml/datasets/phishing+websites)

**Technique : [ Random Forest Classifier ]** The Random Forest Classifier is chosen for its ability to handle high-dimensional data, robustness against overfitting, and efficiency in real-time classification tasks. It provides a reliable method for distinguishing between legitimate and phishing websites based on a variety of features extracted from website characteristics.

**Design and Flow** **:**  The plugin's architecture begins with dataset preparation and model training using Python. The trained classifier is exported and integrated into the Chrome extension, allowing it to analyze URLs on-the-fly as users navigate the web. Upon detection of a potential phishing site, the plugin promptly alerts the user, empowering them to make informed browsing decisions.

|  |
| --- |
| project/  ├── artifacts/  │ ├── backend/  │ │ ├── classifier/  │ │ ├── README.md  │ │ ├── dump.ipynb  │ │ ├── dump.py  │ │ ├── training.ipynb  │ │ └── training.py  │ │  │ ├── dataset/  │ │ ├── README.md  │ │ ├── dataset.arff  │ │ ├── preprocess.ipynb  │ │ └── preprocess.py  │ │  │ └── frontend/  │└── static/ |

Preprocess Dataset :

* This a public phishing site dataset taken from [UCI repository](https://archive.ics.uci.edu/ml/datasets/phishing+websites).
* Download the dataset and save as dataset.arff. The preprocess.py loads the arff file and converts it to numpy array. Then dataset metadata is printed and then dataset is splited into training and testing set with **30%** for testing.  
  Change working directory to /backend/dataset and Run the preprocessor with
* python3 preprocess.py
* Training and testing data \*.npy files are created in the working directory.

**Train and Export RandomForestClassifier :**

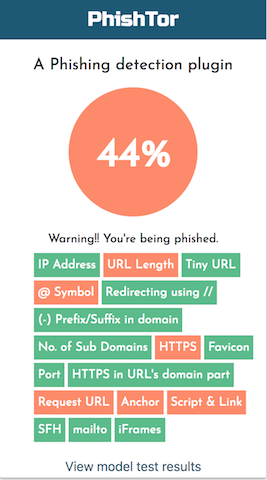
* Before getting to this part, the dataset preprocessing has to be done and the /backend/dataset/ directory should contain test and train .npy files.
* The RandomForestClassifier *(ensemble learner)* is fitted with the training set and then the accuracy and cross validation scores are printed.  
  The parameters of the learned model, such as number of estimators, tree parameters such as thresholds for each estimators are dumped on to a file named classifier.json.
* Change working directory to /backend/classifier and Run
* python3 training.py
* classifier.py is created in the /static directory.  
  Serve this classifier.py over HTTP and update URL in the plugin settings.  
  The same file is also hosted [here](https://raw.githubusercontent.com/picopalette/phishing-detection-plugin/master/static/classifier.json).

Install Plugin :

This requires you to turn on developer mode in chrome extensions. Navigate to chrome://extensions/ and turn on developer mode.

1. You may consider using packed release version of this plugin. Download latest release from [here](https://github.com/picopalette/phishing-detection-plugin/releases) and ignore any warnings. Drag the downloaded .crx file into the chrome://extensions/ page.
2. For unpacked versions, select **load unpacked** and choose the frontend directory of this repository.

Results :



**F1 score:** 0.905

Links to few phishing sites: [PDF](https://github.com/picopalette/phishing-detection-plugin/blob/master/artifacts/url_list.pdf), [PhishTank](https://www.phishtank.com/)

**Conclusion :**

This project demonstrates the feasibility of client-side phishing detection through the "Phishing Site Detector Plugin." By combining machine learning techniques with a commitment to user privacy, the plugin offers an effective solution for enhancing online security. The achieved F1 score of 0.905 validates its accuracy in distinguishing phishing websites, as evaluated against known datasets and references.